National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

NL-339 (r. 88/04)

Canada

THE UNIVERSITY OF ALBERTA

STUDIES IN MATRIX METHODS FOR MODELING MICROSEISMIC EXCITATION OF A LAYERED OIL SAND

by

Douglas Brown

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

IN

GEOPHYSICS

DEPARTMENT OF PHYSICS

EDMONTON, ALBERTA

Fall 1988

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-45600-0

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Douglas G. Brown

TITLE OF THESIS STUDIES IN MATRIX METHODS FOR MODELING MICROSEISMIC EXCITATION OF A LAYERED OIL SAND

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED Fall 1988

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Douglas Brown

PERMANENT ADDRESS:
12518 Ocean Forest Place
White Rock, B.C.
V4A 7G1

DATED Sept 8, 1988

Our poignant adventure, our self sought perils
on a line of unreason to the summit of a superfluous rock,
have no rational or moral justification.

Geoffry Winthrop Young

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled STUDIES IN MATRIX METHODS FOR MODELING MICROSEISMIC EXCITATION OF A LAYERED OIL SAND submitted by DOUGLAS BROWN in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in GEOPHYSICS.

.....

Supervisor

.....

.....

.....

Date September 7, 1988

This thesis is dedicated to my dog Spot

Abstract

Seismic radiation from microseismic events during a steam flood of an oil sand has been observed. These observations may provide useful information if an adequate method for the calculation of synthetic seismograms can be found. Matrix methods combined with Fourier transform techniques may be such a method.

Displacement and stress, directed in the vertical direction, are the only six seismic quantities continuous across boundaries in a horizontally layered elastic structure. The Fourier transforms, with respect to horizontal coordinates and time, of these six quantities satisfy first-order ordinary differential equations. If these quantities are grouped into a vector, these differential equations can be expressed in matrix form, and the stress-displacement vector can be decomposed into components due to upward and downward travelling waves. Zero stress at the free surface, as well as a zero amplitude for upward travelling waves at the the bottom of the stack of layers, are sufficient conditions to allow simultaneous solution of these six differential equations including source contributions by the propagator matrix method. In this way, the transform of displacement and stress can be determined for any depth.

A buried microseismic point source of arbitrary source mechanism can be represented as a seismic moment tensor times spatial derivatives of the Dirac delta function. Volume sources can be expressed as a sum of such point sources.

The calculation of the six by six propagator matrix involves substantial algebra, but can be calculated using computer software that allows symbolic differentiation and algebraic manipulation.

To avoid folding of the time domain seismogram upon itself, a suitably small sampling interval in frequency must be chosen. For realistic geologic models, using frequency and wavenumber bandwidths applicable to seismology, the necessity of choosing a small frequency sampling interval causes the calculation of the transform of the displacement-stress vector to become computationally very strenuous. Consequently, very large amounts of computer time are necessary for the calculation of 3-dimensional seismograms using the propagator matrix method.

Acknowledgements

I would like to express my sincere thanks to my supervisor Dr. Edo Nyland for his help and guidance throughout my graduate studies. His chalk board lectures were a boon only a student of Dr. Nyland could understand.

My greatest indebtedness is to my parents Lloyd and Barbara Brown for always being there and providing support when it was needed.

I would also like to thank the Department of Physics for financial support in the form of Graduate Assistantships.

TABLE OF CONTENTS

CHAPTER		PAGE
1.		1
	Introduction	1
	GLISP	2
	Oil Sand Deposits	5
	Geology of the Athabasca Oil Sands	6
	The Steam Flood	7
	Fracturing in an Oil Sand	8
2.	EXCITATION OF A LAYERED STRUCTURE BY MICROSEISMICITY	11
	Formulation	11
	The Fourier Domain	12
	The Propagator Matrix	13
	The Cayley – Hamilton Theorem	15
	REDUCE2	16
	Substitutions	16
	The Eigenvalues of **A**	16
	Decomposition of the vector **f**	18
	The Eigenvector Matrix	19
	The Source Term	22
	The Transform of the Surface Displacement	23

3. NUMERICAL IMPLEMENTATION OF THE PROPAGATOR MATRIX 24

Calculating the Matrices **A**, **E**, and **P** 24

Evaluation of the Source Term 24

Verification of Results 24

The Discrete Fourier Transform 25

Transformation to the Time - Space Domain 28

4..... 32

Results 32

Investigation of Possible Singularities Encountered in Inverse Fourier Transforming 33

Computational Difficulties in the Evaluation of the Source Contribution 38

Reformulating the Source 41

Another Source Formulation 48

Enforcing a Real **E** 50

Sampling in the Fourier Domain 58

5. CONCLUSIONS..... 61

BIBLIOGRAPHY 64

APPENDIX 1 ALGEBRAIC COMPUTATIONS WITH REDUCE2 66

APPENDIX 2 COMPUTER PROGRAMS..... 86

LIST OF FIGURES

Figure		Page
1.1	The Location of the GLISP site	3
1.2	The Seismometer layout at GLISP During Hydraulic Fracturing	4
1.3	The Seismometer layout at GLISP During Steaming	4
1.4	Types of Grain Contacts	6
1.5	Geology of the GLISP Site	8
1.6	The Mohr stress diagram	9
2.1	Definition of the layer stack. z_n = surface; z_0 = top of half-space; z_s is the depth of the source, but does not necessarily indicate a change in medium properties.	16
3.1	A diagrammatic explanation of my 1-D Fourier Transform approximation for an even valued number of samples	25
3.2	A plot of a 1 Hz and a 4 Hz sine waves sampled (squares) at 5 Hz	28
4.1	Real Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed)	33
4.2	Imaginary Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed)	34
4.3	Branch Cuts and contour path in the complex *p*-plane	36

4.4 Real Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 39

4.5 Imaginary Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 40

4.6 Real Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 44

4.7 Imaginary Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 45

4.8 Real Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 46

4.9 Imaginary Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 47

4.10 Real Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 51

4.11 Imaginary Part of results of an East-West section for Model 1 (NW, NP, NQ = number of w's, p's, and q's summed) 52

4.12 Real Part of results of an East-West section for Model 2 (NW, NP, NQ = number of w's, p's, and q's summed) 53

4.13 Imaginary Part of results of an East-West section for Model 2 (NW, NP, NQ = number of w's, p's, and q's summed) 54

4.14 Results of an East-West section for Model 2 (NW, NP, NQ = number of w's, p's, and q's summed) 56

4.15 Results of an East-West section for Model 2 (NW, NP, NQ = number of w's, p's, and q's summed) 57

4.16 Results of an East-West section for Model 2. Traces beyond the Nyquist distance have been folded into negative distances (NW, NP, NQ = number of w's, p's, and q's summed) 59

Chapter 1

Introduction

A tar sand is a consolidated or unconsolidated rock that contains bitumen or other heavy petroleum that can not be produced by conventional means due to high specific gravity and viscosity at reservoir conditions [Chilingarian and Yen, 1978]. An oil shale is a fine-grained sedimentary rock that yields oil upon heating [Selly, 1985]. In tar sands, oil occurs in a free form in the pores of host rock, but in oil shales, the oil is contained within the complex structure of kerogen (a disseminated organic compound) from which it must be distilled.

Vast quantities of hydrocarbons are stored in both tar sand and oil shale deposits. Substantial difficulty is encountered in attempting to produce oil from tar sand deposits. This difficulty arises from the very viscous, sticky nature of the heavy oil. Due to the substantially increased structural integrity of the shale matrix, as well as the need to distill the oil from the oil shale, the difficulty in producing oil from oil shales is of an order of magnitude more challenging than that encountered in oil production from oil sand deposits.

Approximately ninety percent of the world's known tar sands occur in Canada and Venezuela. The major deposits are the Peace River, Wabasca, Athabasca, and Cold Lake deposits of central Alberta (919 billion barrels of heavy oil), the reserves on Melville Island, NWT, and the Orinoco tar belt, Venezuela (700 billion barrels) [Burger, 1978]. The Melville Island sands are not well studied, but may rate as the third largest tar sand after Athabasca and Orinoco [Phizackerley and Scott, 1978]. The heavy oil in these deposits amounts to nearly 2 trillion barrels of oil, and that is nearly as much heavy oil as the world's total discovered medium and light gravity oil in place [Demaison, 1977]. These deposits are becoming increasingly important as the world's known conventional reserves continue to be used up faster than new reserves are being discovered.

At the conditions present in the geologic formations where tar sands are found, the crude bitumen in the tar sand has very high viscosity and is practically immobile. At reservoir conditions (formation temperature ≈ 13 °C), the heavy oil in the Athabasca and Peace River deposits has viscosities of the order of several million centipoise, while the heavy oil at Cold Lake has viscosities of the order of 10^5 centipoise [Flock and Lee, 1977]. The oil of the Orinoco tar belt has viscosities that are generally lower than those found in Albertan tar sands, and some oil is even producable by primary means and thus strictly speaking does not qualify as a tar sand [Chilingarian and Yen, 1978]. High viscosity is responsible for the extraordinary difficulties encountered in producing oil from tar sand deposits. However, the viscosity of these tar sands decreases exponentially as the temperature is raised above formation temperature: at 200°C, viscosity of Athabasca oil has

dropped to approximately 10 cp [Flock and Lee, 1977]. It is this property of oil sand that allows the thermal recovery techniques used to produce heavy oil to work.

The most popular thermal recovery techniques involve the injection of steam or hot water into the oil sand formations to elevate the temperature and thus reduce viscosity and allow production by conventional methods. It is desirable to be able to observe what happens to the steam front as it propagates through the formation. A reliable method for such observations could lead to improved oil extraction techniques as well as more efficient field operations.

Prior to steam flooding, it is common to fracture the oil sand reservoir by injecting water or a gel compound at high pressure. This fracturing might produce observable seismic emissions, and it is expected that useful information about the geometry of the fracture can be derived from the observation of such seismic energy. The determination of the location and propagation direction of the propagating fracture is difficult, but very desirable. It is hoped, that the seismic observation of the fracturing process will yield valuable hints on the location of the leading edge of the induced fracture.

These two procedures provide two separate seismic sources: (1) the seismic emissions resulting from the initial hydraulic fracturing of the formation, and (2) the micro-fracturing induced as the propagating hot steam front meets the cold, brittle formation. Passive seismic monitoring of emissions from hydraulic fracturing and steam flooding has potential as a realistic method to gain information about the formation and the steam flood as thermal recovery operations proceed.

GLISP

In the fall of 1986, members of the Seismological Laboratory at the University of Alberta were involved in an experimental test of the above hypotheses. A passive monitoring of the Gregoire Lake In-Situ Pilot (GLISP) steam flooding was conducted. GLISP is located in the Athabasca Oil Sands approximately 40 km south of Fort McMurray (figure 1.1). A seismic project consisting of a passive monitoring of microseismic emissions during a hydraulic fracture and subsequent steam flood was conducted. The objective of the project was to determine the value of such observations in the monitoring of the performance of such pilots and the data gathered provided the motivation for the theoretical study of this thesis.

The first phase of the experiment consisted of a continuous seismic monitoring of the GLISP field while wells H4 and H5 (figure 1.2) were being fractured. The array of geophones deployed during this phase is shown in figure 1.2. The results of this monitoring was a noisy dataset severely contaminated by fracture pump noise, with no obvious events of discrete fracture observed.

The second phase consisted of a different, smaller array of geophones (figure 1.3) that were operated in an event detection mode while wells H3, H4, and H5 were being

steamed. During this phase, considerable seismic emissions were detected during the first four days of flooding, but there were few events observed after this time. No seismic events were detected which could be attributed to the steaming in the H4 or H5 wells. The majority of the events were located near the H3 well. The hypocentres of these events lie in a northeast/southwest trending line which corresponds roughly with the location of a known channel [Pullin et al, 1987]. The depths of the epicentres are not known accurately enough to make a comparison to channel depth meaningful.

The purpose of this thesis is to investigate a matrix method of modelling 3-dimensional seismic wave propagation from an internal microseismic source in layered media such as that found in the Alberta oil sands. If the observed seismic radiation results

Figure 1.1 The location of the GLISP site

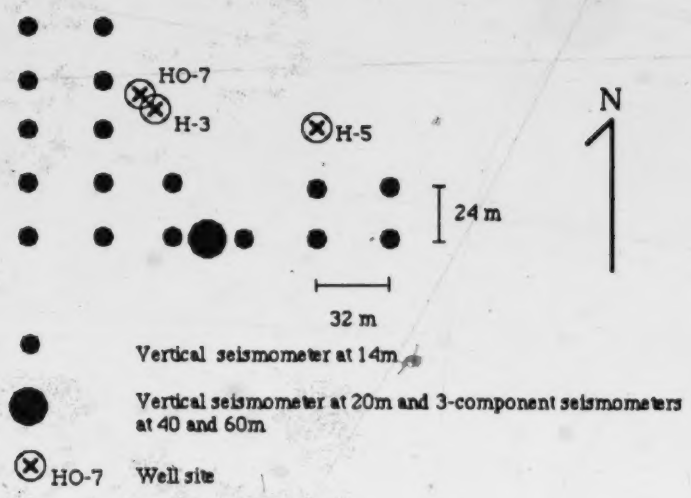

Figure 1.2 The seismometer layout at GLISP during hydraulic fracturing

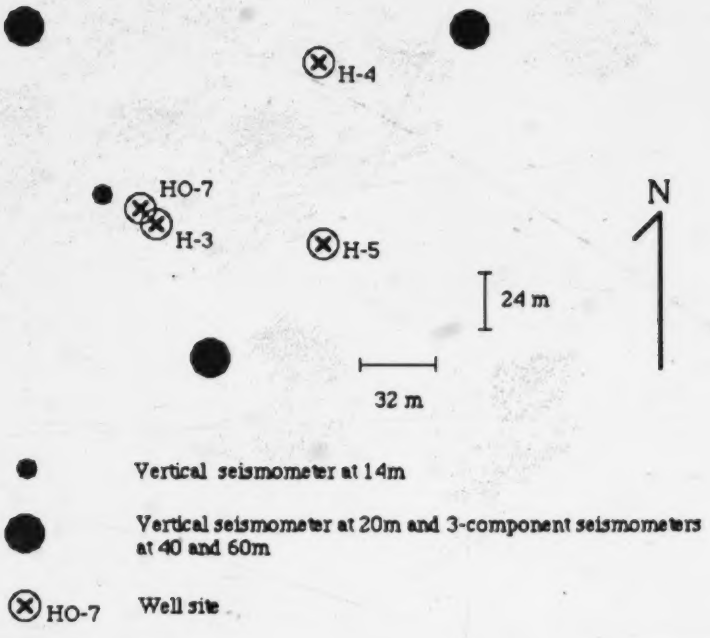

Figure 1.3 The seismometer layout at GLISP during steaming

from fracturing, the results from these matrix methods will correspond to the motions observed at the surface above the tar sand. The comparison of these synthetic results to observed data should lead to better understanding of the source mechanisms, and understanding the source should lead to a better understanding of the processes occurring within the tar sand as steam is injected.

Oil Sand Deposits

For the purpose of this thesis, no distinction shall be made between oil sands and tar sands. Oil sands are deposits of crude oil, that are commonly but not necessarily found in porous sands, and that have API weights of 15° API or heavier, and thus are not recoverable at an economic rate by conventional production methods. The API weight is related to the specific gravity by the formula: API gravity = $141.5/\gamma_0 - 131.5$, where γ_0 is specific gravity at 60° F [Macrides, 1987; Selly, 1985; Demaison, 1977].

The origin of tar sands has long been a topic of debate among geologists. It has been suggested that they are locally generated "immature oil", however there has not been a satisfactory solution proposed to the problem of expelling such highly viscous and sticky material from source rocks and then emplacing it in sand reservoirs.

The presently accepted concept, resulting from significant advances in organic geochemistry, is that tar deposits are the result of water washing and bacterial degradation of in-place light and medium gravity crude oils. Water washing removes the more water soluble light hydrocarbons, while bacterial degradation preferentially removes the normal paraffins, resulting in an increase in density and sulfur content [Baily et al, 1973; Demaison, 1977].

All of the world's major oil sand deposits occur in similar geologic settings. Common to all deposits are moderately rich source beds widespread over large areas, excellent gathering and focused drainage systems into paleo-deltas, very long migration distances, and the predominance of regional stratigraphic factors in both the size and sites of accumulations [Demaison, 1977].

In situ densities of oil sand have been observed to be as high as 2.25 gm/cc, with 2.1 gm/cc typical. This corresponds to a void ratio of ~ 0.40, where the void ratio of a uniform subangular sand is expected to be ≥ 0.50. Oil sands are also known to exhibit shear strengths unexpectedly high for an unconsolidated sand, while demonstrating very low tensile strength. The matrix of oil sand consists of over 90% quartz, with minor amounts of feldspar, muscovite, chert and clay minerals. The explanation of oil sand's compactness and high shear strength lies in the internal arrangement of the sand grains.

Grains in a common sand experience only tangential grain-to-grain contacts, whereas grains in oil sand exhibit mainly "long" and "concavo-convex" contacts (figure 1.4). This structure is also known as interpenetrative or "locked" structure, and it is responsible for both the low void ratios and high shear strengths observed in oil sands [Harris and

Sobkowicz, 1977].

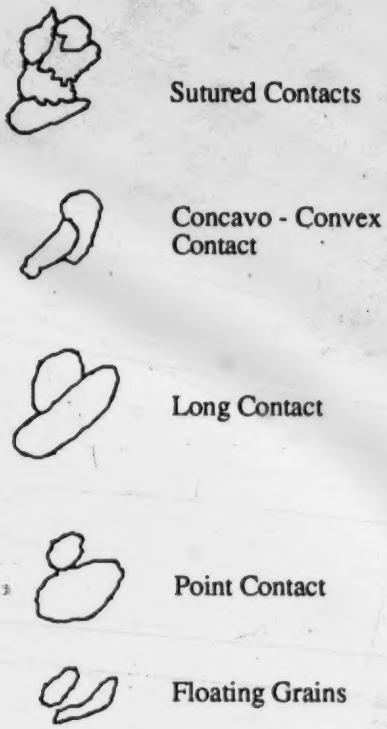

Figure 1.4 Types of grain contacts
(modified from Harris and Sobkowicz, 1977)

The Geology of the Athabasca Oil Sands

The Athabasca oil sands occupy an area of north-eastern Alberta over 320 km long and 150 km across. Within this area, the McMurray Formation contains an estimated 625 billion barrels of oil, making it the world's largest known single hydrocarbon deposit.

The heavy oil of the Athabasca oil sands, as well as the Peace River, Wabasca and Cold Lake oil sands, is contained within the Lower Cretaceous Manville Group. The Manville Group is composed of the Grand Rapids, Clearwater, and McMurray Formations and lies unconformably on Devonian aged carbonate rocks of the Winterburn, Woodbend, or Beaverhill Lake Formations and is unconformably overlaid by marine sediments of the mid-Cretaceous Period [James and Oliver, 1977; Carrigy, 1959]. The sedimentary rocks

of the Cretaceous Period were derived from clastic materials originating in the Canadian Shield lying east of Alberta and from the building Cordillera to the west. The sedimentary material was transported from these two source regions to the Western Canadian Sedimentary Basin by fluvial processes where they were laid down in continental deltaic and marine environments [Macrides, 1987].

Essentially all of the heavy oil of the Athabasca Oil Sands occurs within the McMurray Formation. The thickness of the McMurray is related to relief on the mid-Cretaceous unconformity, with maximum values of about 90 metres. The McMurray Formation consists of predominantly quartz sands, but is complexly interbedded with a sequence of sandstones, siltstones, mudstones, and thin coals. An overall fining upward trend reflects the gradual transition from a fluvial to a deltaic sedimentary environment. The overlying Clearwater Formation consists of mudstones and siltstones deposited during a widespread transgression of the Cretaceous boreal sea. It attains thicknesses of up to 70 m. At the base of this formation, a cherty, glauconitic, fine-grained sandstone called the Wabiskaw Member may be locally present. Approximately 5% of the Athabasca Oil Sands are contained within the Wabiskaw Member . The Grand Rapids Formation consists of fine to medium-grained feldspathic sandstones, laminated sequences of siltstone and sandstone, and thin coals, and is about 105 metres thick [James and Oliver, 1977; Dusseault, 1977].

Using well logs from wells H-4 and H0-7 (fig. 1.2) a seismic model for the GLISP location was generated. These seismic parameters and their geologic interpretation are shown in figure 1.5.

The Steam Flood

Steam flooding is applicable to low energy, high viscosity oil deposits. Low energy refers to reservoir conditions where even if the viscosity of in-situ oil is reduced by heating of the reservoir, there may not be enough drive provided by reservoir pressure, gravity, solution gas drive, etc. to produce present oil. In this case a steam flood is necessary, where the steam not only heats the heavy oil, but also provides the energy to drive the oil out of the formation [Farouq Ali, 1974].

In a steam flood, there are often several injector wells surrounding a central producer well. In the case of GLISP, there were three injector wells about one producer well. High temperature steam is injected at high pressure (near lithostatic pressure) into the injector wells and the softened oil is produced from the central producer well.

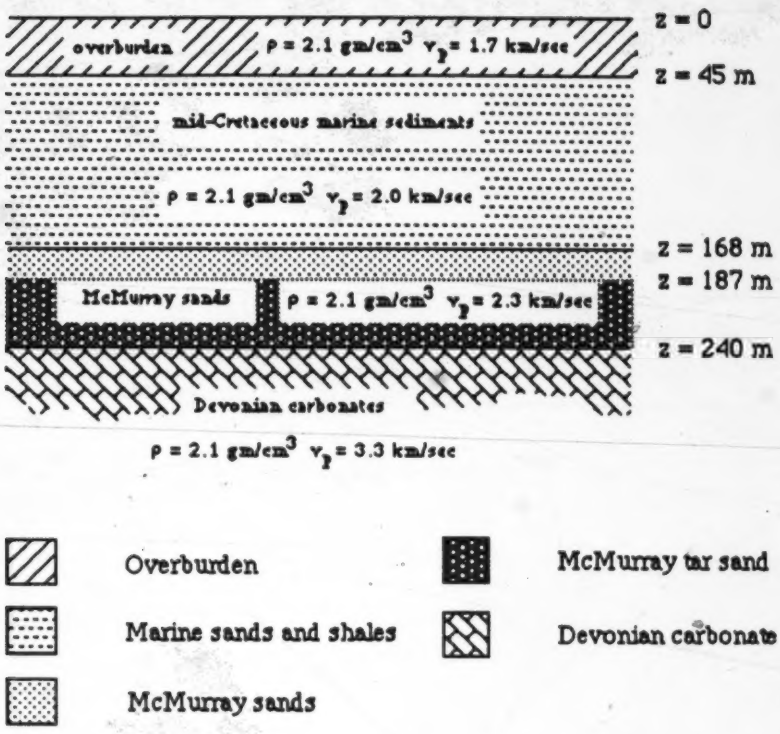

Figure 1.5 Geology of the GLISP site, and the seismic parameters used in modelling its response.

Fracturing in an Oil Sand

It has been established that oil sands display significant shear strength. As previously mentioned, this is a consequence of the nature of the grain contacts, and does not contradict the observation of near zero tensile strength. The question of whether, due to the inherent weak structure of oil sand, the pre-steam flood "fracturing" actually causes seismically observable radiation might reasonably be asked. It also remains to be demonstrated whether or not a steam flood can generate the required shear stresses to cause fracturing and thus generate observable seismic emissions.

Oil sand reservoirs are high porosity sandstones. They have little or no grain-to-grain cohesion, near zero tensile strength, and show considerable dilatancy and non-linear behavior when subjected to changes in stress. Oil sand does not fail by brittle fracture. General dilatancy of ~ 60 - 80 % of pre-yield strain is followed by further dilatancy confined to a narrow shear band [Dusseault and Rothenburg, 1988].

Experimental and field evidence indicate that Alberta oil sands are in a state of stress, and upon the injection of high pressure steam into the formation, failure by fracture is expected [Settari and Raisbeck, 1981]. Dusseault and Rothenburg [1988] contend that during hydraulic fracturing, where injection pressure exceeds maximum principle stress (σ_3), shear bands will develop in advance of fractures. Also, it is stated that steam injection ($0.9\sigma_3 \leq$ injection pressure $< \sigma_3$) of low permeability or weak reservoirs will lead to shear slip along planes inclined to principle stress directions.

The point at which a material under stress fractures is dependent on confining pressure, rock composition, temperature, etc. It is most convenient to analyze the conditions necessary for fracture using a Mohr stress diagram in 2 dimensions. Figure 1.6 shows a diagram of shear stress (τ) versus normal stress (σ). The semi-circle (one half of the Mohr circle) indicates stress conditions within the rock. For a plane making an angle of α with σ_1 (minor principal stress) $\tau = 1/2(\sigma_1-\sigma_3)\cos(2\alpha)$ and $\sigma = 1/2(\sigma_1+\sigma_3) + 1/2(\sigma_1-\sigma_3)\sin(2\alpha)$. The diagonal line labelled "Mohr failure envelope" represents criteria for failure, and is dependent on rock composition and temperature. Whenever a point on the Mohr circle intersects the Mohr failure envelope, failure occurs on the two planes inclined at angles of $\pm\alpha$ to σ_1. The maximum shear stress occurs at $\alpha = 45°$, but failure occurs at some angle less than this depending on the material and the stress state.

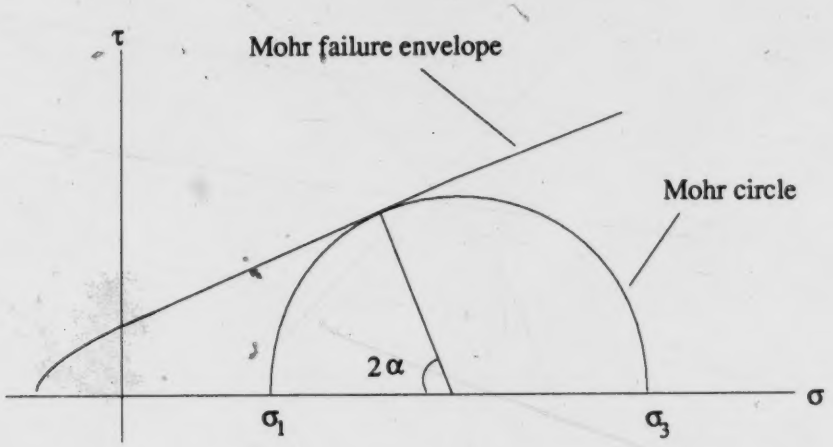

Figure 1.6 A Mohr stress diagram

By injecting high pressure steam into an oil sand, the internal pore pressure is raised, and this in turn counteracts the external pressure and reduces the effective normal stress σ. If this is unaccompanied by thermal or chemical processes of significance it corresponds to moving the Mohr circle left toward the failure envelope. Once the steam injection has produced a sufficiently large pore pressure, the circle contacts the failure envelope at some

point, and fracture occurs. It isn't at all obvious that this fracturing occurs instantaneously, as this conceptual model presumes no pre fracture creep, and no history dependence of the fracture criteria.

Using Griffith crack theory for a penny-shaped crack in an elastic medium, Sneddon [1951] demonstrates that shear stress is generated in the neighborhood of such a crack. In a 2-dimensional vertical cut through a 3-dimensional medium containing a horizontal crack, there are lines of maximum shear stress above and below the fracture plane at 45° to this plane near the edge of the fracture. The maximum shear stress is found just outside the leading edge of the crack. Two mechanisms could be involved in propagating such a crack. Simple parting under pressure is the standard explanation (this requires the injection pressure to exceed lithostatic pressure), but a coalescence of shear failures just ahead of the crack tip is also possible. Despite the fact that any fracturing of an oil sand can not be considered to have occurred in an elastic medium, the Griffith crack model may provide a reasonable first order description of the pre-steam flood fracturing.

Early in the steam flood monitoring at GLISP, numerous good events locatable to the GLISP site were observed, and shear failure in oil sand seems the most likely source. Conventional wisdom states that during a steam flood, any fracturing induced would be controlled by the in situ stress field. In the Gregoire Lake area, the minor principal stress direction is vertical to depths below the bottom of McMurray Formation. No heavy oil is found below the McMurray in the Athabasca Oil Sands. Fractures are known to form normal to the minor principal stress direction, and hence horizontal fracture planes, or fracture planes inclined a few degrees to the horizontal, are expected in Athabasca Oil Sands at Gregoire Lake [Dusseault, 1977; Settari and Raisbeck, 1981].

Anisotropy within the tar sand can be expected to markedly effect the propagation of the steam induced fracture, but Shell Canada Resources Ltd. was able to induce and propagate horizontal fractures in the McMurray formation at a depth of 60 m [Settari and Raisbeck, 1981]. If one accepts the doubtful premise that the in situ stress field still dominates in the altered conditions within a steam flood, then it would follow that any fracturing induced by steam flooding oil sands in the Gregoire Lake area will propagate in nearly horizontal fracture planes. This may be a reasonable assumption, at least initially, as although heating significantly reduces the strength of oil sand, the time necessary to initiate and propagate fractures is small relative to the time necessary to heat an appreciable fraction of the formation [Settari and Raisbeck, 1981]. It is difficult to predict the angle of planes of shear failure that would form during failure of oil sand under steaming or hydraulic fracturing. Theory will be developed for an arbitrary source description, but tested only for the simple case of slip on a horizontal plane so that physical symmetries will be readily observable in simulation results.

Chapter 2

Excitation of a Layered Structure by Microseismicity

Formulation

My aim is to calculate the particle displacement at the surface of a medium that consists of *n* homogeneous layers over an infinite half-space due to a point source described by a moment tensor located inside the layer stack.

Two relationships are sufficient to describe the propagation of plane seismic waves in an elastic homogeneous, isotropic medium: (1) the equilibrium equation:

$$\sigma_{ij,j} = \rho \frac{\partial^2 u_i}{\partial t^2} + g_i \tag{2.1}$$

and (2) a statement of Hooke's law:

$$\sigma_{ij} = \lambda \delta_{ij} u_{l,l} + \mu(u_{i,j} + u_{j,i}) \tag{2.2}$$

where λ and μ are Lame's constants, σ_{ij} is stress in the i-direction on the j-plane, u is displacement, and δ_{ij} is the kronecker delta. g_i is the body force equivalent of a point source function at $\zeta = (x_s, y_s, z_s)$.

For slip on a planar fault surface Σ, the body force equivalent for a point source at position ζ and time t_s can be expressed as

$$g_p(\zeta,t) = -\iint_{\Sigma} [u_i(\zeta,t_s)] c_{ijpq} \nu_j \frac{\partial}{\partial x_q} \delta(x-\zeta) d\Sigma$$

$$= -M_{pq}(t) \frac{\partial}{\partial x_q} \delta(x-\zeta) \tag{2.3}$$

where $x = (x,y,z)$, $\delta(x - \zeta) = \delta(x-x_s)\delta(y-y_s)\delta(z-z_s)$, δ is the Dirac delta function, x_s is the postion of the source, $[u_i]$ indicates displacement across the fault plane from Σ_+ to Σ_-, c_{ijpq} are elastic constants satisfying $\tau_{ij} = c_{ijpq} e_{pq}$ with τ and e denoting stress and strain tensors respectively, ν is unit normal to the surface element $d\Sigma$ pointing from Σ_- to Σ_+, and M_{pq} is the *pq*th component of the source moment tensor [Aki and Richards, 1980, equations (3.5) and (3.23)].

The Fourier Domain

Throughout the problem, I will be working in coordinates Fourier transformed with respect to the x- and y- coordinate directions and with respect to time. This Fourier transform has the form

$$\mathbf{f}(\omega,p,q,z) = \int\int\int_{-\infty}^{\infty} \mathbf{f}(t,x,y,z)\, e^{\sqrt{-1}\,(px + qy - \omega t)}\, dq\, dp\, d\omega \tag{2.4}$$

where p and q are horizontal wavenumbers, and ω is angular frequency.

I will now define the vector **f**

$$\mathbf{f}(z) = \begin{bmatrix} u(z) \\ v(z) \\ w(z) \\ \sigma_{xz}(z) \\ \sigma_{yz}(z) \\ \sigma_{zz}(z) \end{bmatrix}$$

where u, v, and w are displacements in the x-, y-, and z- directions. The six components of **f** are the only quantities that are continuous across a boundary in a horizontally stratified elastic earth model. As I will be working exclusively in the Fourier domain, I will often write $\mathbf{f}(\omega,p,q,z)$ as $\mathbf{f}(z)$ where the dependence on ω, p, and q is implied. For plane waves in the Fourier domain, $\mathbf{f}(t,x,y,z) \Rightarrow \mathbf{f}(z)$, where $\mathbf{f}(z)$ depends on x, y, and t only via the factor $e^{\sqrt{-1}\,(px + qy - \omega t)}$. Thus, derivatives with respect to transformed variables take an especially simple form. For example:

$$\int_{-\infty}^{\infty} \frac{\partial}{\partial x} \mathbf{f}(t, x, y, z)\, e^{\sqrt{-1}\,px}\, dx = -\sqrt{-1}\; p\, \mathbf{f}(t, p, y, z) \tag{2.5}$$

Consequently, for plane waves in the transformed domain, equations (2.1) and (2.2) can be combined in a form that relates the displacement-stress vector **f** to first-order derivatives, with respect to z only, of **f**. I formulate this as

$$0 = \mathbf{A}_L \frac{d\mathbf{f}(z)}{dz} + \mathbf{A}_R \mathbf{f}(z) + \mathbf{s}$$

where **s** is a 6×1 vector contsisting of the p, q, and ω Fourier transform of the g_i's from (2.1). This leads to a relation of the form:

$$\frac{d\mathbf{f}(z)}{dz} = \mathbf{A}(z)\mathbf{f}(z) + \mathbf{g}(z) \tag{2.6}$$

where

$$\mathbf{g}(z) = -\mathbf{A}_L^{-1} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g_1 \\ -g_2 \\ -g_3 \end{bmatrix}$$

and $\mathbf{A} = -\mathbf{A}_L \mathbf{A}_R$.

The Propagator Matrix

Aki and Richards (1980) define a propagator matrix $\mathbf{P}(z,z_0)$ as

$$\mathbf{P}(z,z_0) = \mathbf{1} + \int_{z_0}^{z} \mathbf{A}(\zeta_1)\, d\zeta_1 + \int_{z_0}^{z} \mathbf{A}(\zeta_1) \int_{z_0}^{\zeta_1} \mathbf{A}(\zeta_2)\, d\zeta_2 d\zeta_1 + \ldots \tag{2.7}$$

If for the moment, I neglect the source term, I can write (2.6) as

$$\frac{d\mathbf{f}(z)}{dz} = \mathbf{A}(z)\mathbf{f}(z) \tag{2.8}$$

The propagator satisfies equation (2.8)

$$\frac{d}{dz}\mathbf{P}(z,z_0) = \mathbf{A}(z)\,\mathbf{P}(z,z_0). \tag{2.9}$$

An important property of the propagator matrix is:

$$\mathbf{f}(z) = \mathbf{P}(z,z_0)\,\mathbf{f}(z_0). \tag{2.10}$$

If **f** is known at some depth z_0, this vector can thus be propagated to any depth z with the matrix $\mathbf{P}(z,z_0)$ via equation (2.9).

If there exists a stack of layers i=1, 2, ... , n each of which has a top at $z = z_i$, I can propagate the vector **f** from some level z_0 (say the top of layer 1) to a depth z by:

$$\begin{aligned}\mathbf{f}(z) &= \mathbf{P}(z,z_0)\,\mathbf{f}(z_0) \\ &= \mathbf{P}(z,z_{n-1})\,\mathbf{P}(z_{n-1},z_{n-2})\ldots\mathbf{P}(z_2,z_1)\,\mathbf{P}(z_1,z_0)\mathbf{f}(z_0)\end{aligned} \tag{2.11}$$

where z is in the *n*th layer.

This approach can also be modified to deal with source effects. A solution of (2.6) is given by

$$\mathbf{f}(z) = \int_{z_0}^{z}\mathbf{P}(z,\zeta)\mathbf{g}(\zeta)d\zeta + \mathbf{P}(z,z_0)\mathbf{f}(z_0) \tag{2.12}$$

[Aki and Richards, 1980].

For the case of a homogeneous layer, **A**(z) is constant independent of depth z within the layer, and **P** takes on the particularly simple form:

$$\mathbf{P}(z,z_0) = e^{(z-z_0)\mathbf{A}} \tag{2.13}$$

(see Aki and Richards, 1980 for details).

The exponential of a matrix $(z-z_0)\mathbf{A}$, that may or may not have distinct eigenvalues, is defined in general by

$$e^{(z-z_0)\mathbf{A}} = \frac{1}{2\pi\sqrt{-1}}\oint_C \frac{e^Z}{Z\cdot 1 - (z-z_0)\mathbf{A}}\,dZ \tag{2.14}$$

where the contour C must enclose all of the eigenvalues of **A** [Geortzel and Tralli, 1960]. If I make a substitution for the complex variable Z:

$$X = Z/(z-z_0)$$

I can write equation (2.14) as

$$e^{(z-z_0)A} = \frac{1}{2\pi\sqrt{-1}} \oint_C \frac{e^{(z-z_0)X}}{X \cdot 1 - A} \, dX \tag{2.15}$$

The Cayley – Hamilton Theorem

The Cayley - Hamilton theorem is a theorem of advanced linear algebra that provides an alternate and possibly superior method of calculating the propagator matrix. This discussion is included for completeness sake only, as I did not become aware of it and its value until after I had calculated **P** by (2.14).

The Cayley - Hamilton theorem states that any matrix satisfies its own characteristic equation. That is, if λ is an eigenvalue of a matrix **A** of order n, then the characteristic equation is

$$f(\lambda) = \left| \lambda 1 - A \right| = \lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_0 = 0$$

and

$$F(A) = A^n + a_{n-1}A^{n-1} + \ldots + a_0 I = 0$$

where **I** is the identity matrix. If **A** is degenerate, but semi-simple (has n linearly independent eigenvectors), this can be shortened to

$$F(A) = \sum_{i=0}^{s} a_i A^i = 0$$

where s is the number of distinct eigenvalues. I wish to calculate the exponential of my matrix **A**

$$e^A = \sum_{j=0}^{\infty} \frac{A^j}{j!}$$

However, my matrix **A** is semi-simple, and has 4 distinct eigenvalues, and therefore it follows from the Calyley -Hamilton theorem that every power of **A** greater than 3 can be expressed in terms of lower powers of **A**. and thus I can write

$$e^{\mathbf{A}} = c_0 + c_1\mathbf{A} + \ldots + c_3\mathbf{A}^3$$

It can be shown that there is a unique solution for the coefficients c_i, and their calculation only involves inverting a 4 by 4 non-singular matrix [Deif, 1982].

REDUCE2

The analytical derivation of the propagator matrix for an arbitrary isotropic medium using (2.11) and (2.15) leads to substantial algebra. REDUCE2 [Hearn, 1982] is a LISP-based computer language that allows symbolic algebraic manipulation and differentiation. I used REDUCE2 to do the algebraic computations necessary to determine the matrices **A** and **P**.

Substitutions

Substitutions simplify the algebra, and make many quantities dimensionless. The substitutions I made, as expressed for REDUCE2 where the quantity on the right-hand side of the arrow is substituted for that on the left-hand side, are:

$$p^2 + q^2 \Rightarrow k^2$$

$$\lambda \Rightarrow \alpha^2\rho - 2\mu \qquad\qquad \mu \Rightarrow \beta^2\rho$$

where k is horizontal wave slowness.

The Eigenvalues of A

Solving equations (2.1) and (2.2) simultaneously using REDUCE2, I produced the matrix **A**. **A** is a 6 x 6 matrix and has 4 distinct eigenvalues:

$$\lambda_1 = \sqrt{k^2 - \omega^2/\alpha^2}$$

$$\lambda_2 = \sqrt{k^2 - \omega^2/\beta^2}$$

$$\lambda_3 = \lambda_2$$

$$\lambda_4 = -\lambda_2$$

$$\lambda_5 = -\lambda_2$$

$$\lambda_6 = -\lambda_1$$

Here, as elsewhere in this thesis, the eigenvalues are ordered from largest to smallest. Also, for the remainder of this thesis, z shall be considered to decrease downward. λ_1 and λ_6 represent the vertical wavenumber of up- and down- going P-waves, while λ_2 , λ_3 and λ_4, λ_5 represent the vertical wavenumber of up- and down-going S-waves respectively. λ_2 and λ_4 are repeated due to the degeneracy of the S-waves: SH and SV waves have the same velocity in an isotropic medium.

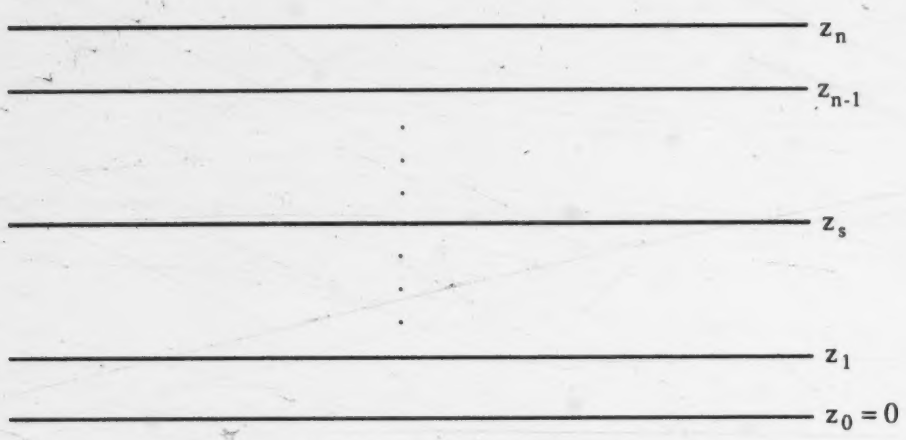

Figure 2.1 Definition of the layer stack. z_n = surface; z_0 = top of half-space; z_s is the depth of the source, but does not necessarily indicate a change in medium properties.

The next step is to calculate the propagator matrix using equation (2.15). The inverse of (**X1**- **A**) can only be singular at the eigenvalues of **A**. Thus, the contour integral in equation (2.15) amounts to calculating the residues at λ_1, λ_2, λ_4,and λ_6: λ_1 and λ_6 are simple poles, while λ_2 and λ_4 are poles of order 2.

Decomposing the vector f

In the absence of sources ($\mathbf{g} = 0$), the differential equation (2.6) can be solved by a different method for homogeneous media. In this case, a solution to this equation is:

$$\mathbf{f} = \mathbf{v}_\alpha e^{\lambda_\alpha (z - z_{ref})} \tag{2.16}$$

where z_{ref} is an arbitrary reference level, $\mathbf{v}_\alpha$ is an eigenvector of $\mathbf{A}$, and λ_α is the associated eigenvalue with no summation over α. As no generality is lost, I will always take the bottom of our stack as reference level ($z_{ref} = 0$). $\mathbf{A}$ is a 6x6 matrix, and I can find 6 eigenvalues and 6 linearly independent eigenvectors ($\alpha = 1,...,6$). Thus, $\mathbf{f}$ can be expressed as:

$$\mathbf{f} = \mathbf{F}\mathbf{w} \tag{2.17}$$

where $\mathbf{w}$ is a vector of constants weighting the columns of $\mathbf{F}$ since the eigenvectors of $\mathbf{A}$ are nonunique to within an arbitrary constant. It follows from (2.16) that the matrix $\mathbf{F}$ can be decomposed as

$$\mathbf{F} = \begin{bmatrix} \mathbf{v}_1, & \mathbf{v}_2, & \ldots, & \mathbf{v}_6 \end{bmatrix} \begin{bmatrix} e^{\lambda_1 z_n} & 0 & \ldots & 0 \\ 0 & e^{\lambda_2 z_n} & 0 & 0 \\ . & . & \ldots & . \\ . & . & & . \\ 0 & \ldots & 0 & e^{\lambda_6 z_n} \end{bmatrix}$$

$$= \mathbf{E}\,\Lambda \tag{2.18}$$

where $\lambda_i \geq \lambda_{i+1}$, and the $\mathbf{v}_\alpha$'s are ordered accordingly. Therefore I can write

$$\mathbf{f} = \mathbf{E}\,\Lambda \begin{bmatrix} \mathbf{w}^{\uparrow} \\ \mathbf{w}^{\downarrow} \end{bmatrix} \tag{2.19}$$

where $\mathbf{w}^{\uparrow}$ and $\mathbf{w}^{\downarrow}$ are 3-component vectors that can be interpreted as the amplitudes of the up- and down- going waves. Given the above decomposition and my definition of the Fourier transform (2.4), it can be observed that **f** propagates via the factor $e^{\sqrt{-1}\,(px + qy + \lambda z - \omega t)}$. It follows from this, that real positive real eigenvalues correspond to upgoing waves, and negative real to downgoing, since z is decreasing downwards and the phase of a propagating wave must be a constant. Imaginary eigenvalues can be interpreted as upward and downward attenuating waves.

The Eigenvector Matrix

I can generate the vector **f** from the P-wave potential $\phi = e^{\sqrt{-1}\,(px + qy + \eta_\alpha z - \omega t)}$ for up-going waves, where $\eta_\alpha = \sqrt{\omega^2/\alpha^2 - k^2}$, as it is known that the displacement arising from ϕ is given by

$$\mathbf{u} = \nabla\phi \tag{2.20}$$

ϕ generated in this way forms the first column of the matrix **E** to within a scalar multiple, which is acceptable because **E** is non-unique to within a scalar multiple. The multiple of the sixth column of **E** can be formed by the same procedure for down-going waves ($\eta_\alpha = -\eta_\alpha$). The middle four columns of **E** can be generated by constructing **f** from S-wave potentials

$$\psi_{SV} = (q, -p, 0)\, e^{\sqrt{-1}\,(px + qy + \eta_\beta z - \omega t)} \tag{2.21}$$

$$\psi_{SH} = (0, 0, 1)\, e^{\sqrt{-1}\,(px + qy + \eta_\beta z - \omega t)} \tag{2.22}$$

where **u** due to ψ is given by

$$\mathbf{u} = \nabla \times \psi \tag{2.23}$$

and $\eta_\beta = \sqrt{\omega^2/\beta^2 - k^2}$ [Aki and Richards, 1980, problem 5.8]. I produced these S-wave potentials essentially by inspection given that $\nabla \bullet \psi = 0$, $\psi_{SV} \times \psi_{SH} = 0$ and $\mathbf{u}_{SH} = (u,v,0)$. This is acceptable given our freedom of a scalar multiple. The resulting eigenvector matrix is

$$\mathbf{E} = \begin{bmatrix} \sqrt{-1}p & \sqrt{-1}p\eta_\beta & \sqrt{-1}q & \sqrt{-1}q & -\sqrt{-1}p\eta_\beta & \sqrt{-1}p \\ \sqrt{-1}q & \sqrt{-1}q\eta_\beta & \sqrt{-1}p & \sqrt{-1}p & -\sqrt{-1}q\eta_\beta & \sqrt{-1}q \\ \sqrt{-1}\eta_\alpha & -\sqrt{-1}k^2 & 0 & 0 & -\sqrt{-1}k^2 & -\sqrt{-1}\eta_\alpha \\ -2\mu p\eta_\alpha & \mu p(k^2-\eta_\beta^2) & -\mu q\eta_\beta & \mu q\eta_\beta & \mu p(k^2-\eta_\beta^2) & 2\mu p\eta_\alpha \\ -2\mu q\eta_\alpha & \mu q(k^2-\eta_\beta^2) & \mu p\eta_\beta & -\mu p\eta_\beta & \mu q(k^2-\eta_\beta^2) & 2\mu q\eta_\alpha \\ \frac{-\lambda\omega^2}{\alpha^2} - 2\mu\eta_\alpha^2 & 2\mu k^2\eta_\beta & 0 & 0 & -2\mu k^2\eta_\beta & \frac{-\lambda\omega^2}{\alpha^2} - 2\mu\eta_\alpha^2 \end{bmatrix} \tag{2.24}$$

This matrix was verified to be a eigenvector matrix of **A** by checking that $\mathbf{E}^{-1}\mathbf{A}\mathbf{E} = \Lambda$, where Λ is a diagonal matrix of the eigenvalues of **A**.

I define the bottom of the stack of layers to be z_0=0, and the surface to be the top of the n'th layer at z_n (see figure 2.1); also, the source is at z_s with z_- is an infinitesimal distance below z_s, and z_+ is an infinitesimal distance above z_s. It follows that (2.12) can be writen as

$$\begin{aligned} \mathbf{f}(z_+) &= \mathbf{P}(z_+,z_-)\,\mathbf{f}(z_-) + \Delta \\ &= \mathbf{f}(z_-) + \Delta \end{aligned} \tag{2.25}$$

since $\mathbf{P}(z_+,z_-) = \mathbf{1}$, where Δ is the contribution due to the source as z increases from z_- to z_+ and is defined by equation (2.12). It follows that I can write

$$\begin{aligned} \mathbf{f}(z_n) &= \mathbf{P}(z_n,z_+)\,\mathbf{f}(z_+) \\ &= \mathbf{P}(z_n,z_+)\,\mathbf{P}(z_-,0)\,\mathbf{f}(0) + \mathbf{P}(z_n,z_+)\,\Delta \\ &= \mathbf{P}(z_n,0)\,\mathbf{E}(0)\,\mathbf{w}(0) + \mathbf{G} \end{aligned} \tag{2.26}$$

since $\mathbf{P}(z_n,z_+)\mathbf{P}(z_-,0) = \mathbf{P}(z_n,0)$, $\mathbf{f}(0) = \mathbf{E}(0)\Lambda(0)\mathbf{w}(0)$ from (2.19), and $\Lambda(0) = 1$. I am justified in using an eigenvector decomposition of $\mathbf{f}$ at $z = 0$ as there are no sources in the infinite half-space.

If the vector $\mathbf{f}$ is grouped into two 3-component vectors of displacement ($\mathbf{u}$) and stress (σ), equation (2.12) can be expressed as

$$\begin{bmatrix}\mathbf{u}\\ \sigma\end{bmatrix}_{z=z_n} = \mathbf{P}(z_n,z_0)\begin{bmatrix}\mathbf{u}\\ \sigma\end{bmatrix}_{z=z_0} + \mathbf{G}(z) \tag{2.27}$$

Also, it is known that at the surface ($z=z_n$) the stress must vanish ($\sigma=0$), and at the bottom of the stack of layers ($z=0$) there is no up-going energy only down-going ($\mathbf{w}^{\uparrow}=0$), and therefore I can replace (2.22) by

$$\begin{bmatrix}\mathbf{u}\\ 0\end{bmatrix}_{z=z_n} = \mathbf{P}(z_n,0)\,\mathbf{E}(0)\begin{bmatrix}0\\ \mathbf{w}^{\downarrow}\end{bmatrix}_{z=0} + \mathbf{G}(z) \tag{2.28}$$

This represents 6 equations in 6 unknowns (the three components of $\mathbf{w}^{\downarrow}$, and the three components of $\mathbf{u}$), and hence a unique solution exists for the transform of the surface displacement $\mathbf{u}(z_n)$ arising from $\mathbf{g}$.

If the matrices are expressed in terms of 3 by 3 submatrices, equation (2.23) can be written as

$$\mathbf{f}(z_n) = \begin{bmatrix}\mathbf{u}(z_n)\\ 0\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{11}(z_n,0)\,\mathbf{P}_{12}(z_n,0)\\ \mathbf{P}_{21}(z_n,0)\,\mathbf{P}_{22}(z_n,0)\end{bmatrix}\begin{bmatrix}\mathbf{E}_{11}(0)\,\mathbf{E}_{12}(0)\\ \mathbf{E}_{21}(0)\,\mathbf{E}_{22}(0)\end{bmatrix}\begin{bmatrix}0\\ \mathbf{w}^{\downarrow}(0)\end{bmatrix} + \begin{bmatrix}\mathbf{G}_1(z)\\ \mathbf{G}_2(z)\end{bmatrix}$$

$$= \begin{bmatrix}\mathbf{P}_{11}(z_n,0)\,\mathbf{P}_{12}(z_n,0)\\ \mathbf{P}_{21}(z_n,0)\,\mathbf{P}_{22}(z_n,0)\end{bmatrix}\begin{bmatrix}\mathbf{E}_{12}(0)\\ \mathbf{E}_{22}(0)\end{bmatrix}\mathbf{w}^{\downarrow}(0) + \begin{bmatrix}\mathbf{G}_1(z)\\ \mathbf{G}_2(z)\end{bmatrix} \tag{2.29}$$

and therefore I can write

$$\begin{bmatrix}\mathbf{u}(z_n)\\ 0\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{11}(z_n,0)\mathbf{E}_{12}(0) + \mathbf{P}_{12}(z_n,0)\mathbf{E}_{22}(0)\\ \mathbf{P}_{21}(z_n,0)\mathbf{E}_{12}(0) + \mathbf{P}_{22}(z_n,0)\mathbf{E}_{22}(0)\end{bmatrix}\mathbf{w}^{\downarrow}(0) + \begin{bmatrix}\mathbf{G}_1(z)\\ \mathbf{G}_2(z)\end{bmatrix} \tag{2.30}$$

The Source Term

I assume initially that the source function described by (2.3) is delta-like:

$$g_i(x,t) = -M_{ij}(t)\frac{\partial}{\partial x_j}\delta(\mathbf{x}-\mathbf{x}_s) = -M_{ij}\delta(t-t_s)\frac{\partial}{\partial x_j}\delta(\mathbf{x}-\mathbf{x}_s)$$

where t_s is the time of the source, $\mathbf{x}_s = (x_s, y_s, z_s)$ and is the position of the source. No generality will be lost if I take the time of the source as origin of our time scale ($t_s = 0$), and $\mathbf{x}_s = (0, 0, z_s)$. Therefore, for i = 1, 2, 3, the Fourier transform of the source function is

$$g_i(z) = \left[\sqrt{-1}\ pM_{i1}\delta(z-z_s) + \sqrt{-1}\ qM_{i2}\delta(z-z_s) - M_{i3}\frac{\partial}{\partial z}\delta(z-z_s)\right]$$

$$= g_i^{*}\delta(z-z_s) + g_i^{**}\frac{\partial}{\partial z}\delta(z-z_s) \qquad (2.31)$$

An expression for the source term **G** follows directly from (2.12), (2.21), and (2.26):

$$\mathbf{G}(z) = \mathbf{P}(z_n,z_+)\Delta$$

$$= \mathbf{P}(z_n,z_+)\int_{z_-}^{z_+}\mathbf{P}(z_+,\zeta)\mathbf{g}(\zeta)d\zeta$$

$$= \mathbf{P}(z_n,z_+)\int_{z_-}^{z_+}\mathbf{P}(z_+,\zeta)\mathbf{g}^{*}\delta(\zeta-z_s)d\zeta + \mathbf{P}(z_n,z_+)\int_{z_-}^{z_+}\mathbf{P}(z_+,\zeta)\mathbf{g}^{**}\frac{\partial}{\partial\zeta}\delta(\zeta-z_s)d\zeta$$

$$= \mathbf{P}(z_n,z_+)\mathbf{P}(z_+,z_s)\mathbf{g}^{*} - \mathbf{P}(z_n,z_+)\frac{\partial}{\partial\zeta}\mathbf{P}(z_+,\zeta)\Big|_{\zeta=z_s}\mathbf{g}^{**} \qquad (2.32)$$

Where **g** is a 6-vector, and is defined in terms of g_i (i = 1, 2, 3) by (2.6). Since z_+ is just above the fault, the layer from z_+ to z_s can be assumed homogeneous and thus using (2.13), $\mathbf{P}(z_+,\zeta)$ can be expressed as a simple exponential of $(z_+ - \zeta)\mathbf{A}_g$, where $\mathbf{A}_g$ is the **A** matrix in the layer immediately above the source. Therefore (2.32) simplifies to

$$\mathbf{G}(z) = \mathbf{P}(z_n,z_s)\mathbf{g}^{*} + \mathbf{P}(z_n,z_s)\frac{\partial}{\partial\zeta}\left[e^{\mathbf{A}_g(z_+-\zeta)}\right]_{\zeta=z_s}\mathbf{g}^{**} \qquad (2.33)$$

Since z_+ is only infinitesimally larger than z_s, $e^{\mathbf{A}_s(z_+ - z_s)}$ is just the identity matrix and hence the following expression for G is produced

$$\mathbf{G}(z) = \mathbf{P}(z_n,z_s)\mathbf{g}^{*} + \mathbf{P}(z_n,z_s)\,\mathbf{A}_g\,\mathbf{g}^{**} \tag{2.34}$$

The Transform of the Surface Displacement

It follows immediately from (2.25) that:

$$\mathbf{w}^{\downarrow}(0) = -\left[\mathbf{P}_{21}(z_n,0)\mathbf{E}_{12}(0) + \mathbf{P}_{22}(z_n,0)\mathbf{E}_{22}(0)\right]^{-1}\mathbf{G}_2(z) \tag{2.35}$$

And finally, combining (2.25) and (2.29), I get an expression for the surface displacement inyolving only the source term; eigenvalue, eigenvector, and propagator matrices:

$$\mathbf{u}(z_n) = -\left[\mathbf{P}_{11}(z_n,0)\mathbf{E}_{12}(0) + \mathbf{P}_{12}(z_n,0)\mathbf{E}_{22}(0)\right]\left[\mathbf{P}_{21}(z_n,0)\mathbf{E}_{12}(0) + \mathbf{P}_{22}(z_n,0)\mathbf{E}_{22}(0)\right]^{-1}\mathbf{G}_2(z) + \mathbf{G}_1(z) \tag{2.36}$$

Chapter 3

Numerical Implementation of the Propagator Matrix

The next step in the process of producing synthetic seismograms is to implement equation (2.36) on a digital computer.

Calculating the Matrices A, E, and P

I already have analytical expressions for the matrices $\mathbf{A}(z)$, $\mathbf{E}(z)$, and $\mathbf{P}(z,z_0)$, and I need only to supply the quantities that describe the medium as function of depth z: that is, given seismic wave velocities $\alpha(z)$ and $\beta(z)$, density $\rho(z)$, I can specify **A**, **E**, and **P** as a function of temporal and spatial frequencies ω, p, and q. To complete the evaluation of the non source term in (2.36), I need only to specify z_n, the depth from the top of the semi-infinite halfspace to the top of the stack of layers.

Evaluation of the Source Term

Since I have specified the time of source as origin time ($t_s = 0$), the calculation of the source term **G** in (2.36) requires only the depth of the source (0, 0, z_s) and the source moment tensor **M**. Given these two quantities, **G** is easily calculated by simple matrix algebra as specified in (2.34).

Verification of Results

Aki and Richards [1984] define their **f** vector slightly differently than I did. Their definition is as follows:

$$\mathbf{f}(z) = \begin{bmatrix} u(z) \\ v(z) \\ \sqrt{-1}\, w(z) \\ \sigma_{xz}(z) \\ \sigma_{yz}(z) \\ \sqrt{-1}\,\sigma_{zz}(z) \end{bmatrix}$$

Comparing this to our definition of **f** (2.6), you will notice that Aki and Richards have a

multiplicative factor of $\sqrt{-1}$ in the third and sixth rows of **f** as compared to our **f**. If one examines equations (2.6) and (2.9), it is obvious that the effect of this to multiply the third and sixth columns of our **A** and **P** matrices by $-\sqrt{-1}$, and to multiply the third and sixth rows our **A** and **P** matrices by $\sqrt{-1}$ as compared to their matrices. These two together mean that the 3-3 and 6-6 components have a net multiplicative factor of 1.

Aki and Richards give the matrix **A**, **E**, and **P** for the cases of P-SV waves, and for SH waves only. By setting one of my wavenumbers to zero, I in effect decompose my solution into the limiting cases of P-SV and SH waves. For example by setting q=0, I constrain my results to the x-z plane, and can compare the non-zero elements of my matrices directly with the published results of Aki and Richards. My results for **A** and **P** were numerically compared for these limiting cases for the range of ω, p, and q's I am considering. They were consistent to better than four significant figures. I tested **E** by ensuring that it satisfied the equation $\mathbf{E}^{-1}\mathbf{AE} = \Lambda$, where Λ is a diagonal matrix of the eigenvalues of **A**.

The Discrete Fourier Transform

The inverse Fourier Transform $\mathbf{u}(\omega,p,q)$ is defined as

$$\mathbf{u}(t,x,y) = \frac{1}{8\pi^3}\int\int\int_{-\infty}^{\infty}\mathbf{u}(\omega,p,q)\, e^{-\sqrt{-1}(px + qy - \omega t)}\, dq\, dp\, d\omega \qquad (3.1)$$

where p and q are horizontal wave numbers, and ω is angular frequency [Aki and Richards, 1980]. I can write (3.1) as

$$\mathbf{u}(t,x,y) = \frac{1}{4\pi^2}\int\int\int_{-\infty}^{\infty}\mathbf{u}(f,p,q)\, e^{-2\pi\sqrt{-1}\left(\frac{px}{2\pi} + \frac{qy}{2\pi} - ft\right)}\, dq\, dp\, df \qquad (3.2)$$

where $f = \omega/2\pi$ and is frequency in Hertz. If I let $\hat{p} = p/2\pi$ and $\hat{q} = q/2\pi$ (3.2) can be rewritten as

$$\mathbf{u}(t,x,y) = \int\int\int_{-\infty}^{\infty}\mathbf{u}(f,\hat{p},\hat{q})\, e^{-2\pi\sqrt{-1}(\hat{p}x + \hat{q}y - ft)}\, d\hat{q}\, d\hat{p}\, df \qquad (3.3)$$

Since I am trying to model a band limited signal (see below for a complete discussion) I can assume our signal has zero amplitude outside some given range of

frequencies and wavenumbers, and it is therefore reasonable to replace (3.3) by a band-limited approximation

$$u(t,x,y) \approx \int_{-F/2}^{F/2} \int_{-\hat{P}/2}^{\hat{P}/2} \int_{-\hat{Q}/2}^{\hat{Q}/2} \mathbf{u}(f,\hat{p},\hat{q})\, e^{-2\pi\sqrt{-1}(\hat{p}x + \hat{q}y - ft)}\, d\hat{q}\, d\hat{p}\, df \qquad (3.4)$$

and approximating these integrals by sums

$$\mathbf{u}(t_i,x_j,y_k) \approx \sum_{l=\frac{-(L-1)}{2}}^{\frac{L-1}{2}} \sum_{m=\frac{-(M-1)}{2}}^{\frac{M-1}{2}} \sum_{n=\frac{-(N-1)}{2}}^{\frac{N-1}{2}} \mathbf{u}(f_l,\hat{p}_m,\hat{q}_n)\, e^{-2\pi\sqrt{-1}\,(\hat{p}_m x_j + \hat{q}_n y_k - f_l t_i)}\, \Delta\hat{q}\, \Delta\hat{p}\, \Delta f \qquad (3.5)$$

Here I have used Simpson's rule to approximate the triple integral, where the integrals are broken up into L, M, and N rectangles with the height measured at the midpoint of each rectangle. In this way, there are an equal number of sample points of the function on each side of zero p, q, and f, and thus the transformed result must be real if expected symmetry is present (see below for a discussion of symmetry).

If L, M, and N are odd, then there will be an odd number of integer valued l, m, and n's (eg. if N = 5, then n = -2, -1, 0, 1, 2) where l, m, and n = 0 correspond to f, p, and q = 0 respectively. If however, L, M, and N are even, then there will be an even number of non-integer values of l, m, and n (eg if N=8, then n=-7/2, -5/2, ... , 5/2, 7/2) (see figure 3.1). In this later case, l, m, and n are never equal to zero, and thus the zero or DC level of our function is not sampled. Since the propagator matrix is singular for f = 0 , I will always use only odd values of L.

It is somewhat unusual to use non-integer values for the indexes of our summations, but below, following a transformation of our indices, l, m, and n are constrained to be integers regardless of the choice of L, M, and N (see (3.6)).

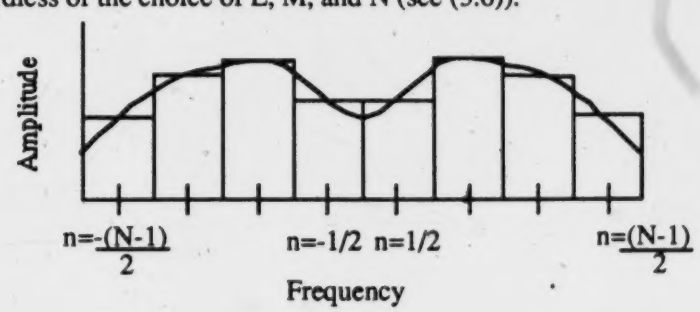

Figure 3.1
A diagrammatic explanation of our 1-D Fourier Transform approximation for an even valued number of samples

I set the temporal sampling interval in the untransformed domain (Δt) to be the inverse of the maximum frequency in our sum:

$$\Delta t = 1/F$$

I also define $f_i = i\Delta f = iF/L$, for $i = -\frac{L-1}{2}, -\frac{L-1}{2}+1, \ldots, 0, \ldots, \frac{L-1}{2}-1, \frac{L-1}{2}$

Similarly,

$$\Delta x = 1/\hat{P} \qquad \Delta y = 1/\hat{Q}$$

$$\hat{p}_m = m\Delta\hat{p} = m\hat{P}/M \qquad \hat{q}_n = n\Delta\hat{q} = n\hat{Q}/N$$

for $m = -(M-1)/2, \ldots, (M-1)/2$ for $n = -(N-1)/2, \ldots, (N-1)/2$

And hence (3.5) becomes

$$u(t_i, x_j, y_k) \approx \sum_{l=\frac{-(L-1)}{2}}^{\frac{L-1}{2}} \sum_{m=\frac{-(M-1)}{2}}^{\frac{M-1}{2}} \sum_{n=\frac{-(N-1)}{2}}^{\frac{N-1}{2}} u\left(\frac{lF}{L}, \frac{m\hat{P}}{M}, \frac{n\hat{Q}}{N}\right) e^{-2\pi\sqrt{-1}\left(\frac{li}{L}+\frac{mj}{M}-\frac{nk}{N}\right)} \frac{\hat{Q}}{L}\frac{\hat{P}}{M}\frac{F}{N}$$

$$= \frac{\hat{Q}}{N}\frac{\hat{P}}{M}\frac{F}{L} \sum_{l=0}^{L-1}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} u\left[\left(l-\frac{L-1}{2}\right)\frac{F}{L}, \left(m-\frac{M-1}{2}\right)\frac{\hat{P}}{M}, \left(n-\frac{N-1}{2}\right)\frac{\hat{Q}}{N}\right] e^{-2\pi\sqrt{-1}\left(\frac{li}{L}+\frac{mj}{M}-\frac{nk}{N}\right)}$$

$$e^{\pi\sqrt{-1}\left[\frac{(L-1)}{L}i + \frac{(M-1)}{M}j - \frac{(N-1)}{N}k\right]} \qquad (3.6)$$

The last term im (3.6) is phase correction that is a result of approximating the integrals (3.4) by short sequences. Obviously, it has a norm of 1.

A straight forward implementation of (3.6) would require $L^2*M^2*N^2$ operations; however, utilizing a Fast Fourier Transform algorithm, (3.6) can be done in NNlog(NN) operations, where NN = L*M*N. This amounts to a substantial computational savings for even moderately large L, M, and N. The IMSL routine FFT3D computes

$$u(i, j, k) = \frac{1}{LMN} \sum_{l=0}^{L-1}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} u(l, m, n)\, e^{-2\pi\sqrt{-1}\left(\frac{li}{L}+\frac{mj}{M}-\frac{nk}{N}\right)} \qquad (3.7)$$

utilizing a FFT algorithm. The calculation of (3.6) was done using FFT3D (3.7), and thus **u** must be ordered properly (ie. as l goes $0 \rightarrow L-1$, f_l goes $-F/2 \rightarrow F/2$) before calling FFT3D, and the appropriate scaling and phase shift factors be applied to the summed values.

Expression (2.4) relates the surface displacement to its transform:

$$\mathbf{u}(\omega,p,q) = \int_{-\infty}^{\infty}\int\int \mathbf{u}(t,x,y)\, e^{\sqrt{-1}\,(px + qy - \omega t)}\, dq\, dp\, d\omega \qquad (2.4)$$

If u(t,x,y) is real, the following observation of symmetry can be made:

$$\mathbf{u}(-\omega,-p,-q) = \int_{-\infty}^{\infty}\int\int \mathbf{u}(t,x,y)\, e^{-\sqrt{-1}\,(px + qy - \omega t)}\, dq\, dp\, df$$

$$= \mathbf{u}^*(p,q,\omega) \qquad (3.8)$$

where * indicates complex conjugate.

If axis in transform space (ω, p, and q) is considered each to be of two sections, one section relating to positive frequency or wavenumber and the other to negative frequency or wavenumber, eight cubes in ωpq - space are defined. If one examines the equations that define the propagator matrix, and hence **u**, it can be seen that only second order derivatives with respect to t exist, and thus only factors of ω^2 appear in **u**, since **u** depends on t only via the factor $e^{(\sqrt{-1}\,\omega t)}$. This means that **u** is symmetric about $\omega = 0$ {ie. $\mathbf{u}(\omega,p,q) = \mathbf{u}(-\omega,p,q)$}. Using the above two symmetry relations, we need only calculate two of the eight cubes to completely define $\mathbf{u}(\omega,p,q)$:

$$\mathbf{u}(\omega,p,q) = \mathbf{u}(-\omega,p,q) = \mathbf{u}^*(\omega,-p,-q) = \mathbf{u}^*(-\omega,-p,-q)$$
$$\mathbf{u}(\omega,-p,q) = \mathbf{u}(-\omega,-p,q) = \mathbf{u}^*(\omega,p,-q) = \mathbf{u}^*(-\omega,p,-q)$$

Transformation to the Time - Space Domain

The above calculations are done in the frequency domain for a specified f, p, and q. Since our source is a spike in time and space, it must contain contributions from all temporal and spatial frequencies. However, in practice the observed seismogram is limited temporally by instrumentation and sampling interval, and spatially by seismometer spacing. This means that the frequency content of a particular observed seismogram is limited to a certain bandwidth dependent on the experiment parameters.

Since we are dealing with a digitally sampled analogue waveform, we need to address the phenomenon of aliasing. We need a minimum of two samples per cycle to define any given waveform. Normally, one samples a waveform in the time domain and then transforms to the frequency domain. In this situation, the largest frequency that can be

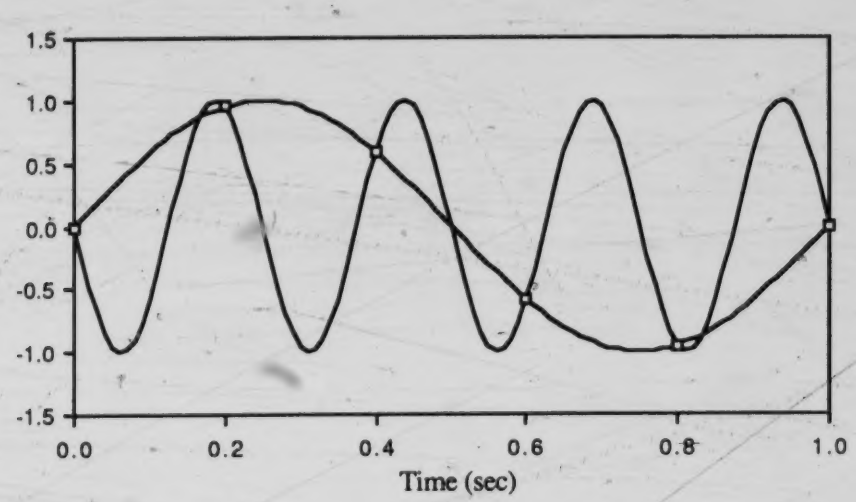

Figure 3.2 A plot of a 1 Hz and a 4 Hz sine waves sampled (squares) at 5 Hz

represented is given by one over twice the sampling interval in time; this is called the Nyquist frequency: $f_N = \frac{1}{2\Delta t}$. If there exists energy at frequencies above the Nyquist frequency, this energy will be "folded back" into the range $0 \rightarrow f_N$ and thus will distort results. Figure 3.2 demonstrates that with a sampling interval of 0.2 sec (f_N = 2.5 Hz), no distinction can be made between a 1 Hz and a 4 Hz wave. As a consequence, any energy at 4 Hz, will be aliased and show up as energy at 1 Hz. The common solution to this problem is to filter the analogue signal to ensure that no energy exists above f_N.

A 2 msec sampling interval in time was used during the seismic monitoring at GLISP. This corresponds to a Nyquist frequency of 250 Hz. At GLISP, the seismic signal was fitered with a low pass filter with a cut off of about 120 Hz before digitizing, and therefore there is not a problem of temporal aliasing.

In a direct analogy to temporal aliasing, one must also consider spatial aliasing. Considering the situation of seismometers in a horizontal plane detecting horizontally travelling energy, waves with a wavelength less than twice the seismometer spacing will not be resolved, and their energy will be aliased back into the range of wavelengths that can be resolved.

At GLISP, the seismometers are in material which has a compressional wave velocity (α) of ~ 2000 m/s. The smallest seismometer spacing in the E/W direction used at GLISP was 32 m (see figure 1.2). This means that we cannot resolve energy horizontally propagating in this direction that has wavelengths less than 64 m. This in turn implies that waves with frequencies > ~ 30 Hz will be spatially aliased (since velocity = frequency × wavelength). For example, spatial aliasing will map energy at ~ 35 Hz to ~ 25 Hz, and thus the observed energy at 25 Hz includes energy from E/W horizontally travelling waves with frequencies of 25, 35 and 85 Hz (assuming no energy above 90 Hz). Similiarly, given a shear wave velocity $\beta \sim 1/\sqrt{3}\,\alpha \sim 1150$ m/s, shear waves with frequencies above ~ 20 Hz will experience spatial aliasing. The spectra of waves detected at GLISP show significant energy below 30 Hz, and since the seismometers are insensitive to these frequencies, spatial aliasing at GLISP is suspected.

Given that vertically incident waves have an infinite apparent wavelength, and are thus not subject to spatial aliasing, the above discussion represents a worst case scenario. Since at GLISP the source is buried a depth of the order of twice the array size directly beneath the seismometer array, there will have to be excitation of surface/interface waves before spatial aliasing will become a problem

In our case, we are in effect sampling in the frequency domain and transforming to the time domain. We will model this by calculating $\mathbf{u}(p,q,f)$ for a finite number of f, p, and q's (L, M, and N respectively) with sampling intervals chosen considering these numbers and the physical dimensions of our model. For example, we need a maximum frequency of 100 Hz to agree with the experiment we are trying to model, and therefore,

$$f_{Nyq} = \frac{1}{2\Delta t} = 90\ \text{Hz}$$

and

$$\Delta f = \frac{1}{T} = \frac{1}{L\Delta t} = \frac{180}{L}$$

and thus

$$F = L\Delta f = 180 = 2f_{Nyq}$$

In (3.6), we sum from -F/2 to F/2 which corresponds to frequencies from $-f_{Nyq}$ to f_{Nyq}.

For p and q, it is the smallest spatial sampling interval that indicates what frequencies can be resolved. The smallest seismometer spacing is 32 metres in the x-direction (E-W) and 24 metres in the y-direction (N-S). The shortest wavelength observable is 2 × seismometer spacing, and therefore,

$$\hat{P} = \frac{1}{2\Delta x} = \frac{1}{64} \qquad \hat{Q} = \frac{1}{2\Delta y} = \frac{1}{48}$$

and

$$\Delta\hat{p} = \frac{\hat{P}}{M} \qquad \Delta\hat{q} = \frac{\hat{Q}}{N}$$

Using the conjugate symmetry relations, the matrix u(f,p,q) will be filled up, and then utilizing the inverse discrete Fourier Transform, u(t,x,y,0) will be calculated.

The effect of a wave with a band-limited frequency content was modelled by defining a zero amplitude for frequencies 0-30 Hz and 90-120 Hz. This amounts to padding u with zeros before inverse Fourier transforming.

Chapter 4

Results

The theory developed in chapters in 2 and 3 was implemented on a digital computer (see appendix 2 for code). The first thing to be noted, is that when calculating all quadrants of the transform of u ($u(\omega,p,q)$), the symmetry predicted by (3.8) is absent. As a consequence, when all of u is calculated explicitly with no enforced symmetry, the displacement in the time-space domain ($u(t,x,y)$) is complex.

For my first synthetic calculations, I chose a simple earth model consisting of a single 400 metre thick layer with seismic velocities α = 2000 m/s, β = 1150 m/s, and with a density of 2100 kg/m^3, over a half-space with α = 2600 m/s, β = 1800 m/s, and ρ = 2400 kg/m^3 The seicmic moment tensor is for slip on a horizontal plane, and thus the only non-zero terms in **M** are $M_{13} = M_{31} = 10^{10}$ N m = seismic moment. The source is burried at a depth of 375 metres.

The velocity of ground motion above microseismically active area is of the order of 10^{-6} m/s for a 10 Hz wave [Aki and Richards, 1980, p. 497]. An earthquake source takes the form of step function. Since I have specified my source as delta-like, I have given the derivative of an earthquake source, so instead of calculating surface displacement, I am calculating surface velocity.

The seismic moment is given by $M_0 = \mu \times$ average slip $\times$ fault area, where μ is shear modulus [Aki and Richards, 1980]. Microseismic faulting with a slip of 1 cm and a fault area of 1 m^2 is reasonable guess for fracturing at GLISP. Given that the source is the top layer, the seismic moment of this source is: $M_0 \sim 10^7$ N m. I have specified a seismic moment of 10^{10}, and thus velocities of the order of 10^{-3} are expected from my calculations.

I calculated u for 40 ω's (20 of these are zeros corresponding to frequencies −30 - 30 Hz, 90 - 120 Hz, and −90 - −120 Hz), 20 p's, and 20 q's, and inverse Fourier transformed via (3.6). The ω's were evenly spaced between −120 and 120 Hz, the p's were evenly spaced between $-1/2\Delta x_s/2\pi$ and $1/2\Delta x_s/2\pi$, and the q's were evenly spaced between $-1/2\Delta y_s/2\pi$ and $1/2\Delta y_s/2\pi$ (where Δx_s and Δy_s are the minimum seismometer spacings in the east-west and north-south directions respectively). The numbers of frequencies calculated was essentially an arbitrary decision, but the cost of computer time was a factor. Since u is symmetric with respect to ω, u only had to be calculated for 10 frequencies (10 frequencies are determined by symmetry and the other twenty are zero).

In the discussion on the discrete inverse Fourier transform in chapter 3, I define $\Delta t = 1/F$, $\Delta x = 1/\hat{P} = 1/(\Delta x_s)/(2\pi)$, and $\Delta y = 1/\hat{Q} = 1/(\Delta y_s)/(2\pi)$. Therefore, the total time represented in a seismic trace will be the number of samples in a trace (40) times the inverse

of the total frequency bandwidth (240 Hz), which is 0.17 sec. Similiarly, since dx_s is 32m, the maximum x value is 4021 m.

The real and imaginary components of a E-W section of the above calculation are plotted in figures 4.1 and 4.2. The produced seismograms are oscillatory with no distinct arrivals. The amplitudes are of the order of 10^{-4} m/s for minimum and maximum x, and of the order of 10^{-5} m/s for intermediate values of x. Traces for y = 0 have amplitudes of zero. These results are unexpected and cleary non-physical. They indicate a need to reexamine the theory.

Investigation of Possible Singularities Encountered in Inverse Fourier Transforming

The following material is based on chaper 6 of Aki and Richards [1980]. In these examples I consider the acoustic or SH problem only. The full elastic problem introduces only algrebraic, not fundamental, complexity.

Given a point source in an infinite homogeneous space with time dependence $e^{-i\omega t}$ (where i is $\sqrt{-1}$), a solution to the inhomogeneous wave equation is given by

$$\phi(\mathbf{x},t) = \frac{1}{R} e^{-i\omega(R/c - t)} \tag{4.1}$$

where $R^2 = x^2 + y^2 + z^2$, and c is wave velocity. Using Fourier transform techniques to express this spherical wave as a sum of plane waves, the Weyl integral is obtained

$$\frac{1}{R} e^{i\omega R/c} = \frac{1}{2\pi} \iint_{-\infty}^{\infty} \frac{e^{i(px + qy - \eta|z|)}}{i\eta} \, dp \, dq \tag{4.2}$$

where $\eta = \sqrt{\omega^2/c^2 - p^2 - q^2}$. This amounts to essentially the same integral I am calculating numerically over horizontal wavenumbers p and q. It is obviously singular for $c = \pm \sqrt{\omega^2/(p^2 + q^2)}$. If there is a zero wave amplitude as $|z| \to \infty$, η must be constrained such that Im $\eta \leq 0$. However, for the range of ω, p, and q's I am considering, and given the seismic velocities of interest, η is always real, and the positive square root is taken, and thus any singular points are avoided.

Transforming (4.2) to cylindrical coordinates, the Sommerfeld integral is produced:

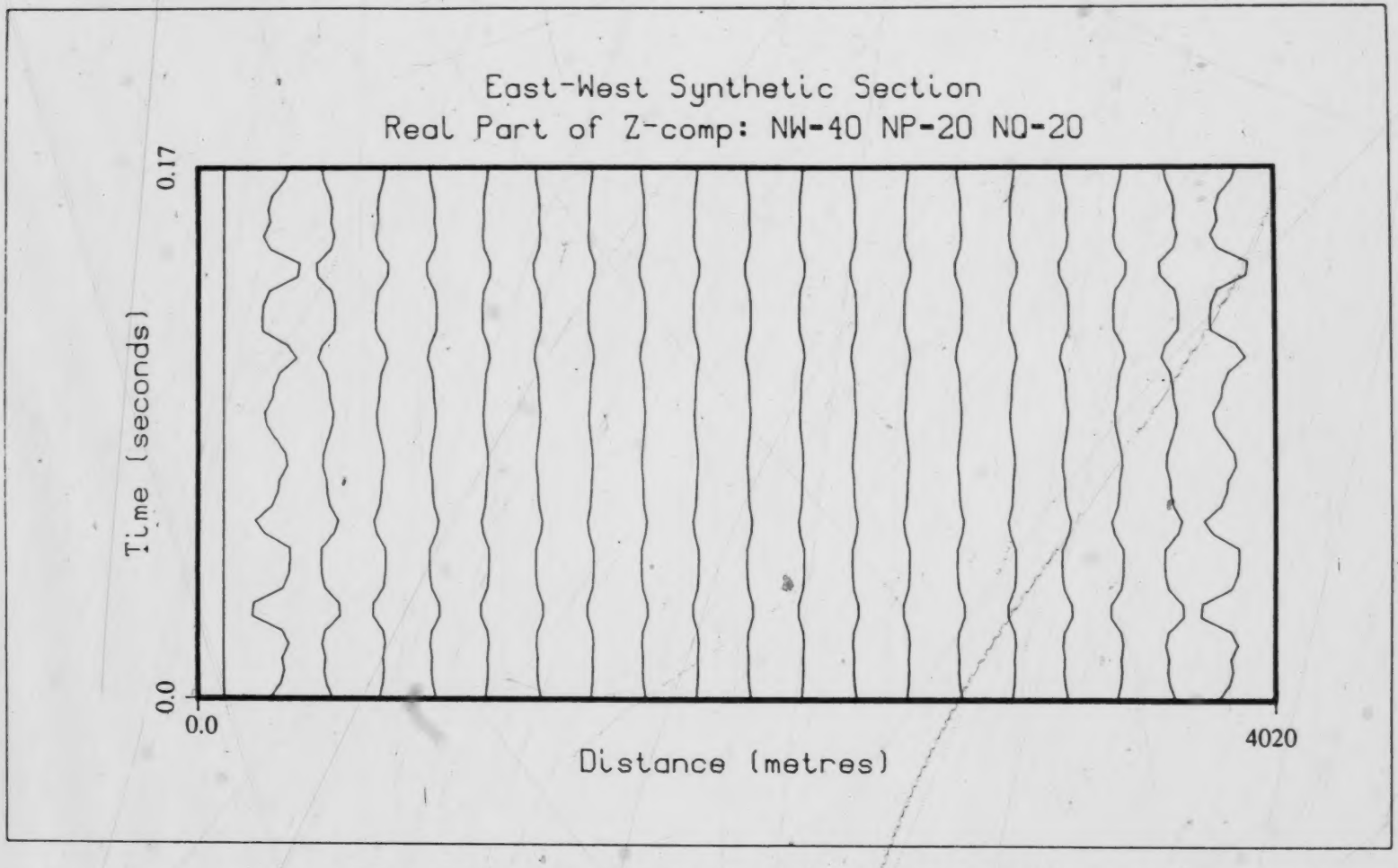

Figure 4.1 Real Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

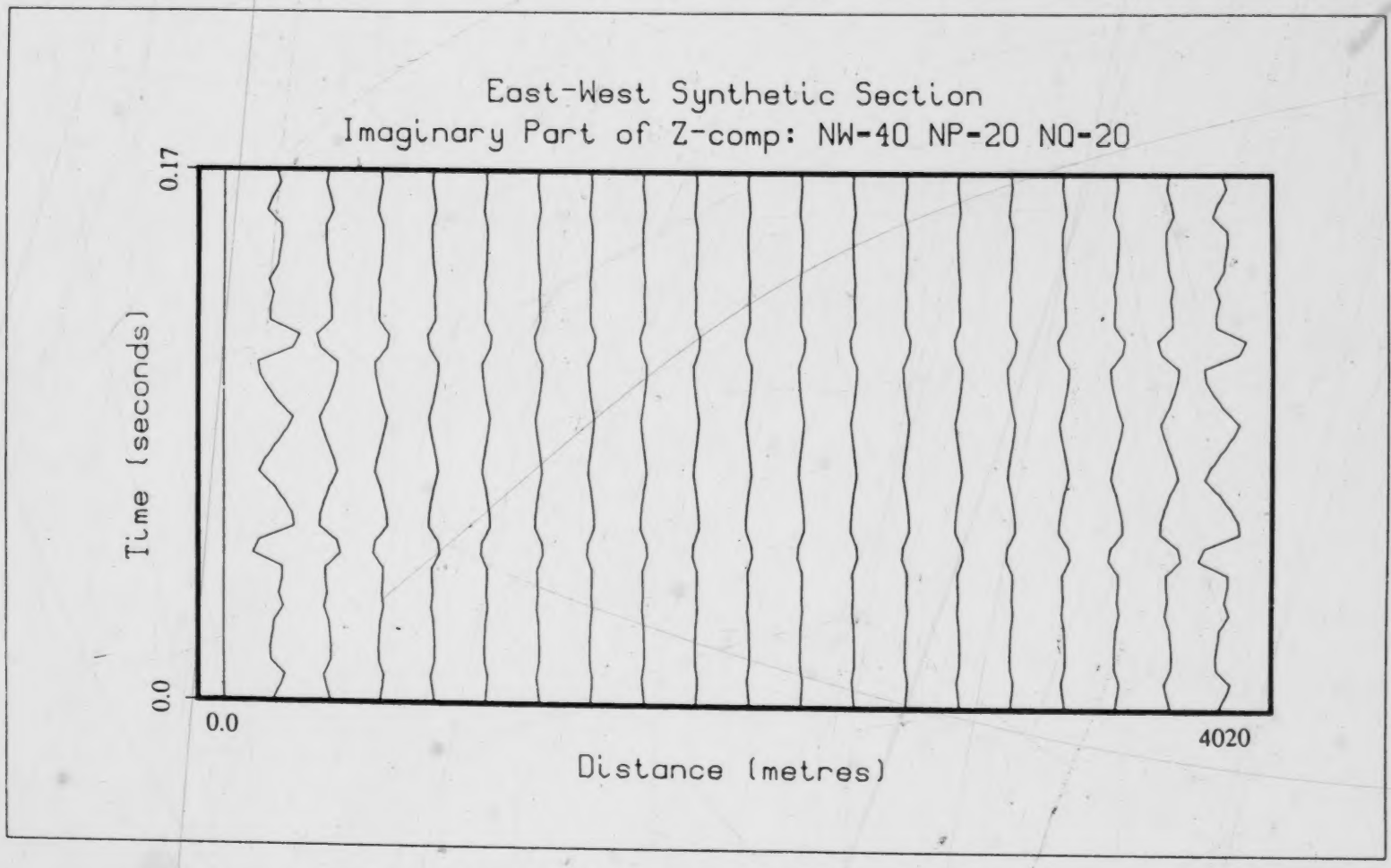

Figure 4.2 Imaginary Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

$$\frac{1}{R} e^{i\omega R/c} = \int_0^\infty \frac{k_r J_0(k_r r)\, e^{-i\eta|z|}}{i\eta} dk_r \tag{4.3}$$

where $k_r = \sqrt{x^2 + y^2}$, $\eta = \sqrt{\omega^2/c^2 - k_r^2}$, and J_0 is the zero order Bessel function. Equation (4.3) represents a decompostion of a sperical wave into a sum of cylindrical waves instead of plane waves.

By making the substitution $k_r = \omega p$, where p is the ray parameter, (4.4) is obtained.

$$\frac{1}{R} e^{i\omega R/c} = i\omega \int_0^\infty \frac{p}{\xi} J_0(\omega p r)\, e^{-i\omega\xi|z|} dp \tag{4.4}$$

where $\xi = \eta/\omega$ and is i times the vertical wave slowness. Just as it was required that Im $\eta \geq 0$, here the choice Im $\xi \geq 0$ must be made. For application of complex analysis, one desires that the integrand of (4.4) be analytic. To this end, it is required that ξ be a single-valued function of p. By dividing the complex p-plane into two Riemann sheets corresponding to Im $\xi > 0$ and Im $\xi < 0$, a branch cut Im $\xi = 0$ joining these two sheets is defined. ξ is single-valued and analytic on each of these sheets. Given Im $\xi = 0$, it follows that $(c^{-2} - p^2)$ is real and non-negative. Thus

$$c^{-2} - (\mathrm{Re}\, p)^2 + (\mathrm{Im}\, p)^2 - 2i(\mathrm{Re}\, p)(\mathrm{Im}\, p) \geq 0.$$

Allowing c to be complex (corresponding to an attenuating medium), with Im c small and positive, it follows that

$$\mathrm{Im}\,(c^{-2}) - 2(\mathrm{Re}\, p)(\mathrm{Im}\, p) = 0$$

and thus

$$(\mathrm{Re}\, p)(\mathrm{Im}\, p) = \varepsilon$$

where $\varepsilon \to 0$ for a perfectly elastic medium. And thus the branch cut is a hyperbola in the p-plane. Additionally, since $\mathrm{Re}(c^{-2} - p^2) \geq 0$, it follows that

$$(\mathrm{Im}\, p)^2 \geq (\mathrm{Re}\, p)^2 - \mathrm{Re}(c^{-2})$$

and the branch cuts are limited to those shown schematically in figure 4.3. These branch cuts must be avoided in the evaluation of the integral (4.4).

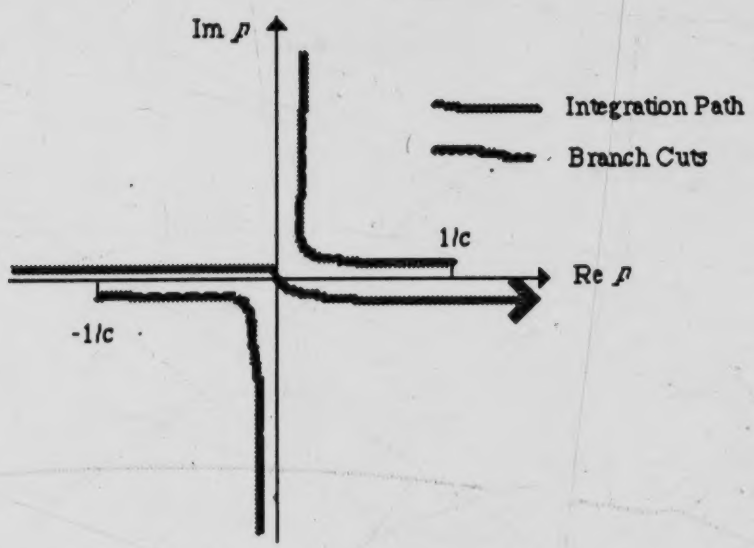

Figure 4.3 Branch cuts and contour path in the complex p-plane

When one considers even simple situations consisting of a few homogeneous layers, the complexity of the problem increases dramatically, and very intricate paths of integration are required. I refer the reader to sections 6.2 and 6.3 of Aki and Richards [1980] for more details.

The important observation to relate to the above discussion, is that I am not free to choose the sign of the η's in my calculations. The signs in **E** are constrained by assigning to $e^{i\eta z}$ the interpretation of either upward and downward travelling waves (real η) or upward and downward attenuating waves (imaginary η). **P** was calculated by finding residues at $\pm\eta_\alpha$ and $\pm\eta_\beta$, and thus the signs of the η's are determined by pole locations. But, more importantly, by numerically calculating band-limited approximations to the integrations, I have avoided the complicating regions ($-1/c < \mathrm{Re}p < 1/c$). By avoiding these regions, I have also neglected the contributions of some types of waves (eg. surface waves).

Computational Difficulties in the Evaluation of the Source Contribution

Since singularities encountered in inverse Fourier transforming are not the cause of my unsatisfactory results, the evaluation of the source term may be a possible source of trouble.

First of all, I am specifying a source that is band-limited in frequency. And thus the frequency spectrum of my source is a boxcar, and when I inverse FFT a sinc(t) behavior is produced. An oscillatory seismogram is to be expected from an oscillatory source. The synthetics in figures 4.1 and 4.2 were recomputed for a non-band-limited frequency band. Frequencies from 0 to 90 Hz were summed. Some results from this calculation are shown in figures 4.4 and 4.5. Although these results have much of the same unexpected characteristics as figures 4.1 and 4.2, they are much less oscillatory in time. They also have similar amplitudes.

Additionally, I have specified the source function as a derivative of a delta function. When I inverse FFT with respect to wavenumbers p and q, I am attempting to numerically integrate an integral having a form

$$\int_{-\infty}^{\infty} ipe^{ipx}\,dx \tag{4.5}$$

(see (2.31)). For (4.5) to be integrable, ip must be square integrable [Sneddon, 1951]. Since ip is in fact not square integrable, (4.5) is divergent. I accepted a band-limited approximation. On close inspection of this calculation, it is clear that this was not wise:

$$\int_{-\infty}^{\infty} ipe^{ipx}\,dx = \lim_{P\to\infty} \int_{-P}^{P} ipe^{ipx}\,dx \tag{4.6}$$

$$= \lim_{P\to\infty} \left[\frac{p}{x} e^{ipx}\Big|_{-P}^{P} - \int_{-P}^{P} \frac{1}{ix} e^{ipx}\,dx \right]$$

$$= \lim_{P\to\infty} \left[\frac{P}{x} e^{iPx} + \frac{P}{x} e^{-iPx} + \frac{1}{x^2} e^{iPx} - \frac{1}{x^2} e^{-iPx} \right]$$

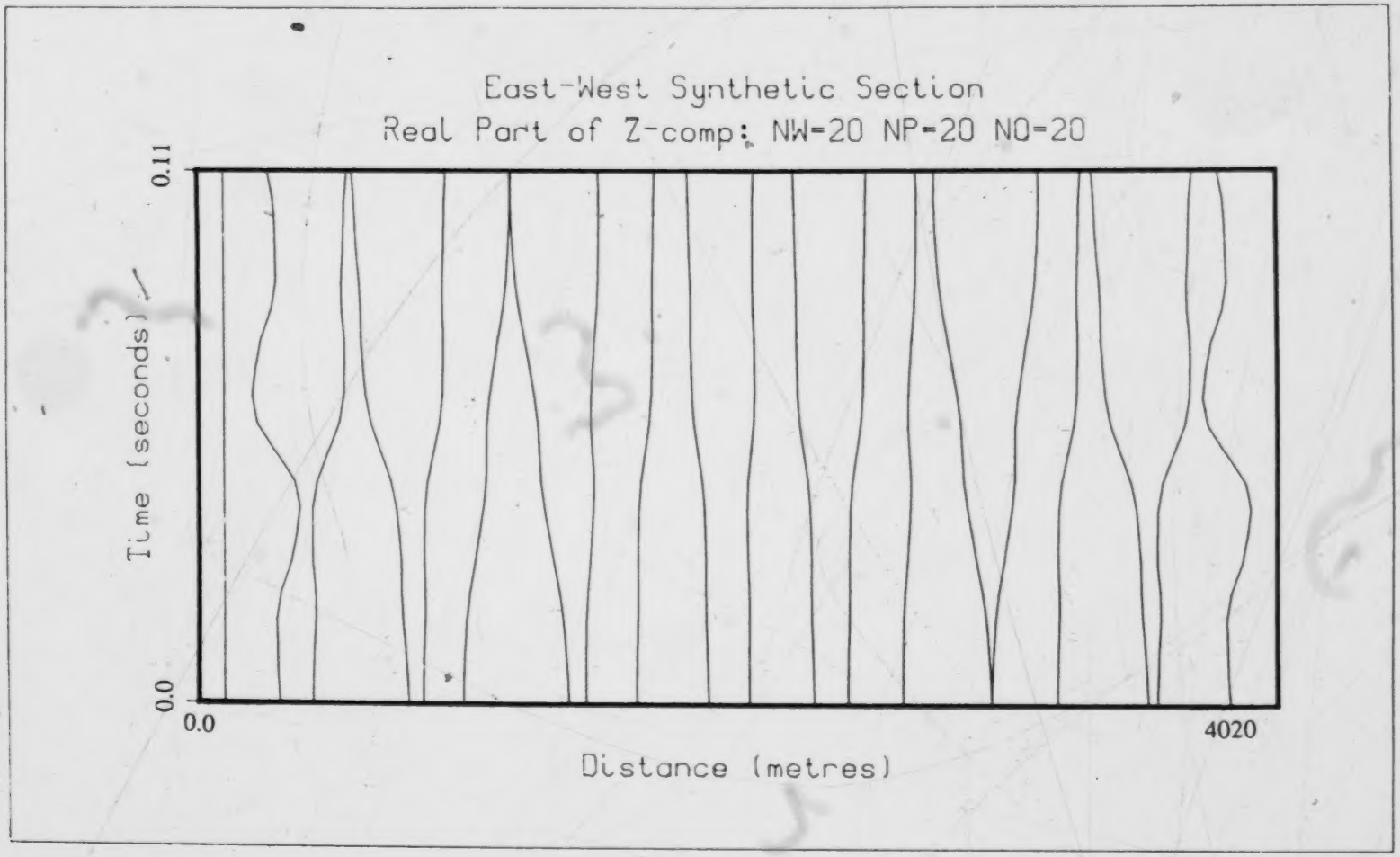

Figure 4.4 Real Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

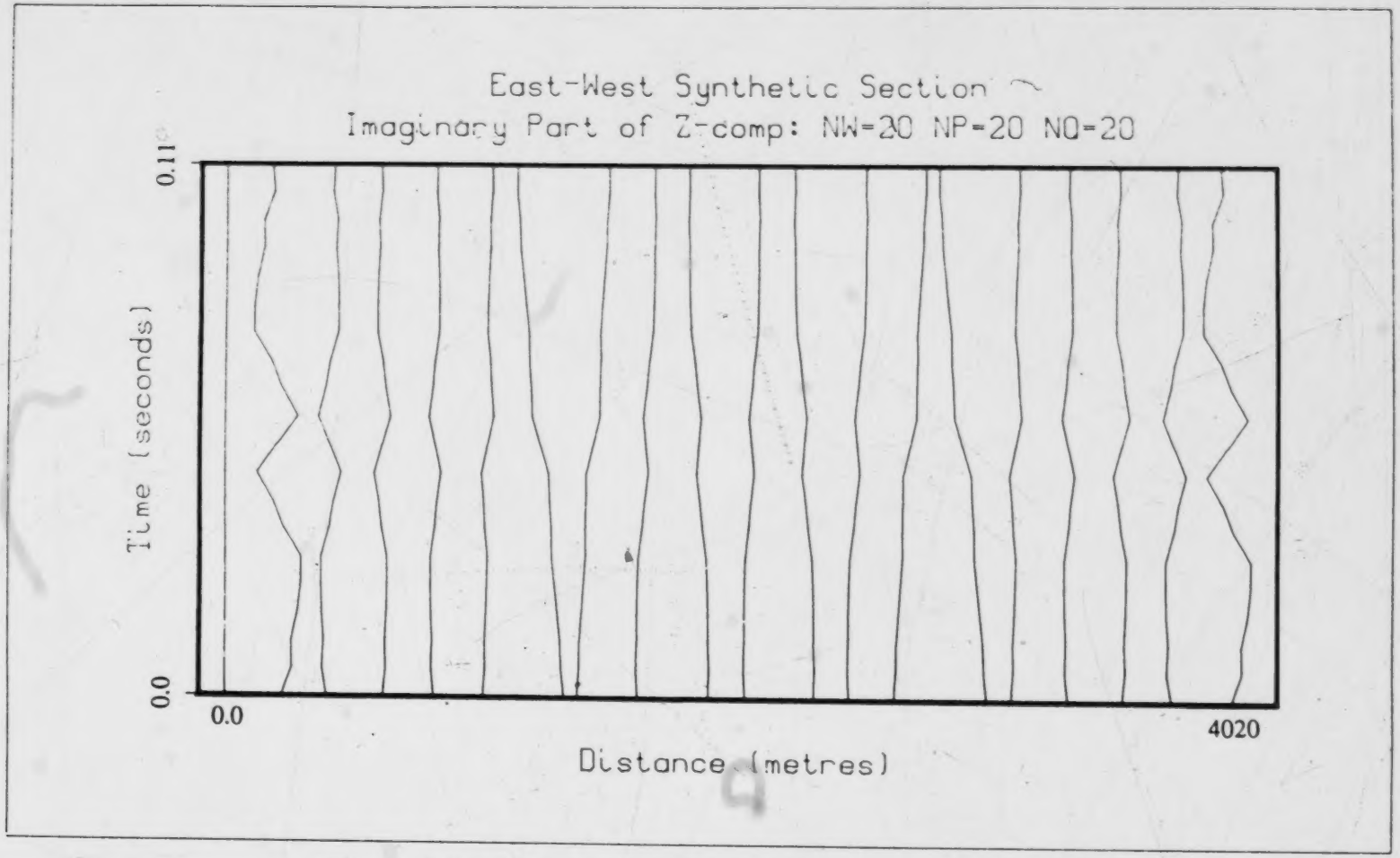

Figure 4.5 Imaginary Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

$$= \lim_{P\to\infty}\left[\frac{2P}{x}\cos(Px) + \frac{2}{x^2}\sin(Px)\right] \qquad (4.7)$$

Equation (4.7) represents an oscillatory function for non-zero P. This function begins to oscillate very rapidly as P gets large. As $x \to 0$, it approaches the derivative of a delta function; that is, approaching from postive x it goes to $+\infty$, and approaching from negative x it goes to $-\infty$.

Reformulating the Source

The problem stems from an inability to take the Fourier transform of the derivative with respect to x, of the delta function $\delta(x)$ (and of $\partial/\partial y$ of $\delta(y)$). A feasible solution might be to approximate the delta function by a function of the form

$$\delta(x) = A\, e^{-\frac{|x|}{a}} \qquad (4.7)$$

where A and a are constants > 0. Given (4.7), I can write

$$\int_{-\infty}^{\infty} \delta(x)\, dx = \int_{0}^{\infty} A\, e^{-\frac{x}{a}} dx + \int_{-\infty}^{0} A\, e^{\frac{x}{a}} dx$$

$$= Aa + \int_{0}^{\infty} A\, e^{-\frac{x}{a}} dx$$

$$= 2Aa \qquad (4.8)$$

[Hodgman et al, 1959]. And, if the constants are chosen such that A = 1/(2a), then

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 1 \qquad (4.9)$$

as required for the Dirac delta function. It follows from (4.7) that the x-derivative of the delta function is

$$\frac{d}{dx}\delta(x) = \begin{cases} \frac{-A}{a} e^{-\frac{x}{a}}, & x \geq 0 \\ \frac{A}{a} e^{+\frac{x}{a}}, & x < 0 \end{cases} \tag{4.10}$$

It can be seen that this definition of $\delta(x)$ solves my problem since

$$\int_{-\infty}^{\infty} \left| \frac{d}{dx}\delta(x) \right| dx = \int_{-\infty}^{0} \left| \frac{A}{a} e^{+\frac{x}{a}} \right| dx + \int_{0}^{\infty} \left| \frac{-A}{a} e^{-\frac{x}{a}} \right| dx$$

$$= \int_{0}^{\infty} 2\frac{A}{a} e^{-\frac{x}{a}} dx$$

$$= 2A \tag{4.11}$$

and therefore the Fourier transform of (4.10) is convergent. Choosing a determines the "width" of the new delta function; that is how far from x = 0 does $\delta(x)$ have a significant non-zero amplitude.

The effect of using a definition for $\delta(x)$ and $\delta(y)$ given by (4.7) is change the source from a point source to one which occupies a finite area in the x-y plane, and this will make some slight modification to the expression of my source.

I now specify my source as

$$\begin{aligned} g_i(x,t) = &-M_{i1}\frac{\partial}{\partial x} Ae^{-\frac{|x|}{a}} Be^{-\frac{|y|}{b}} \delta(z-z_0)\delta(t) \\ &- M_{i2}\frac{\partial}{\partial y} Ae^{-\frac{|x|}{a}} Be^{-\frac{|y|}{b}} \delta(z-z_0)\delta(t) \\ &- M_{i3}\frac{\partial}{\partial z} Ae^{-\frac{|x|}{a}} Be^{-\frac{|y|}{b}} \delta(z-z_0)\delta(t) \end{aligned} \tag{4.12}$$

where A=1/(2a) and B=1/(2b). The Fourier transform of the first term in (4.12) is given

by

$$\int\int\int_{-\infty}^{\infty}\left[-M_{i1}\frac{\partial}{\partial x}Ae^{-\frac{|x|}{a}}Be^{-\frac{|y|}{b}}\delta(z-z_0)\delta(t)\right]e^{i(px+qy-\omega t)}\,dt\,dy\,dx =$$

$$= -M_{i1}\delta(z-z_0)\int_{-\infty}^{\infty}Be^{-\frac{|y|}{b}}e^{iqy}\left[\int_{-\infty}^{0}\frac{A}{a}e^{\frac{x}{a}}e^{ipx}\,dx-\int_{0}^{\infty}\frac{A}{a}e^{\frac{x}{a}}e^{-ipx}\,dx\right]dy$$

$$= -M_{i1}\delta(z-z_0)\int_{-\infty}^{\infty}Be^{-\frac{|y|}{b}}e^{iqy}\left[\frac{A}{a}\int_{0}^{\infty}e^{-\frac{x}{a}}\left[e^{-ipx}-e^{ipx}\right]dx\right]dy$$

$$= -M_{i1}\delta(z-z_0)\int_{-\infty}^{\infty}Be^{-\frac{|y|}{b}}e^{iqy}\left[\frac{-2iA}{a}\int_{0}^{\infty}e^{-\frac{x}{a}}\sin(px)\,dx\right]dy$$

$$= -M_{i1}\delta(z-z_0)\left[\frac{A}{a}\frac{-2ip}{a^{-2}+p^2}\right]\int_{-\infty}^{\infty}Be^{-\frac{|y|}{b}}e^{iqy}\,dy$$

$$= 4M_{i1}\delta(z-z_0)\left[\frac{A}{a}\frac{p}{a^{-2}+p^2}\right]\left[B\frac{q}{b^{-2}+q^2}\right] \qquad (4.13)$$

[Hodgman et al, 1959]. The transform of the second term in (4.12) is found in a similiar way, and thus g_i^* in (2.31) is now given by

$$g_i^* = 4M_{i1}\left[\frac{A}{a}\frac{p}{a^{-2}+p^2}\right]\left[B\frac{q}{b^{-2}+q^2}\right]+4M_{i2}\left[A\frac{p}{a^{-2}+p^2}\right]\left[\frac{B}{b}\frac{q}{b^{-2}+q^2}\right] \qquad (4.14)$$

with g_i^{**} unchanged from (2.31).

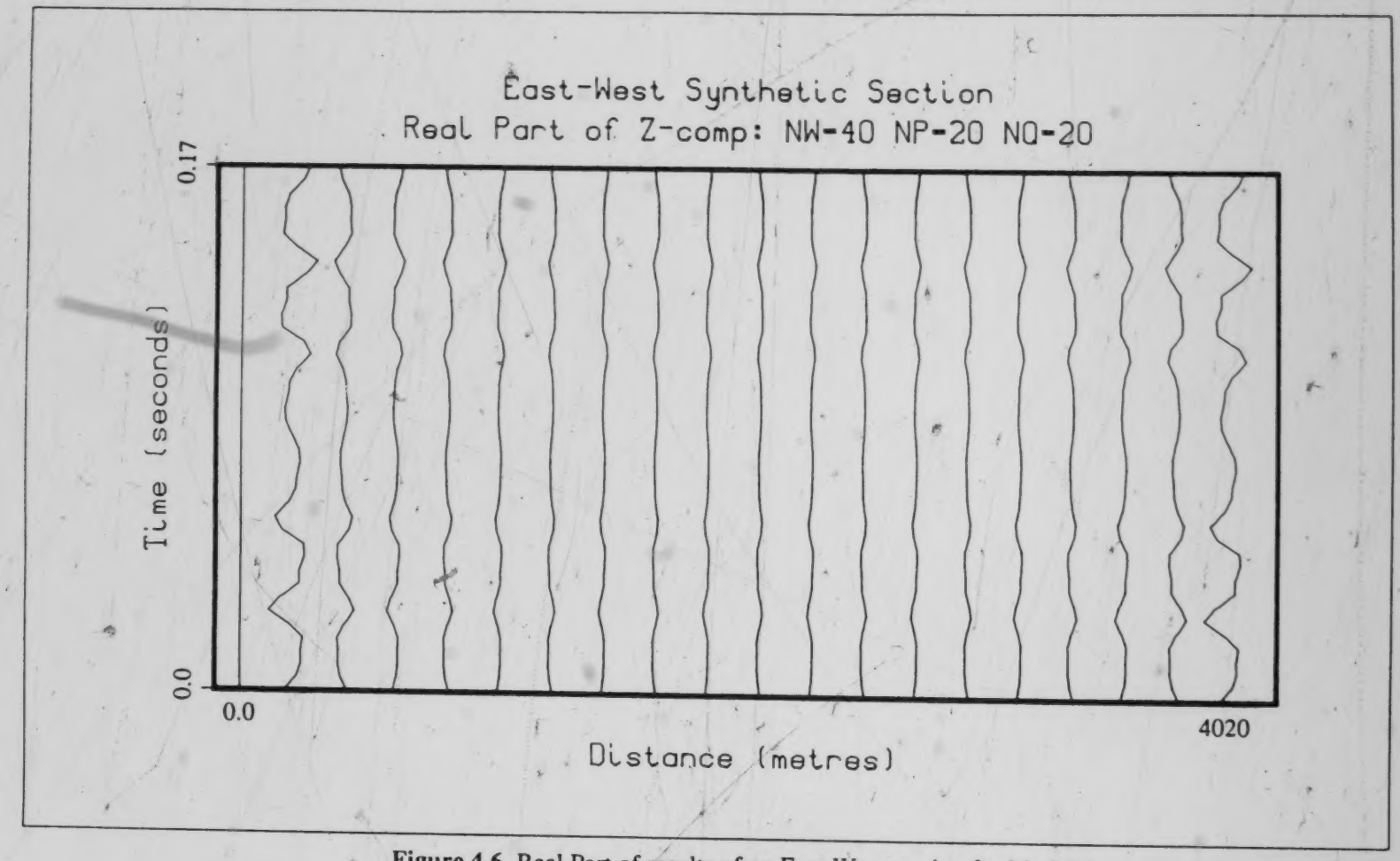

Figure 4.6 Real Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

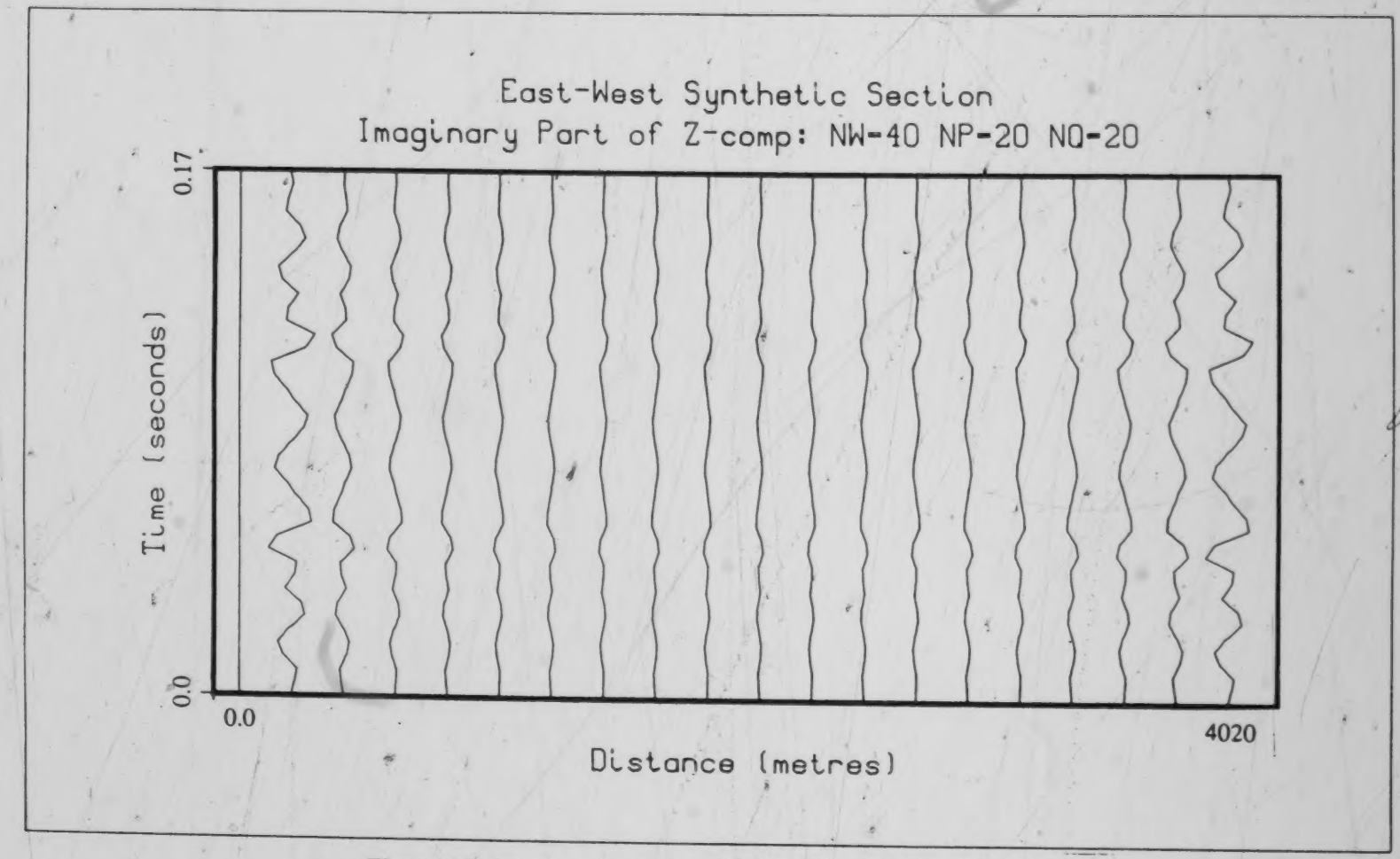

Figure 4.7 Imaginary Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

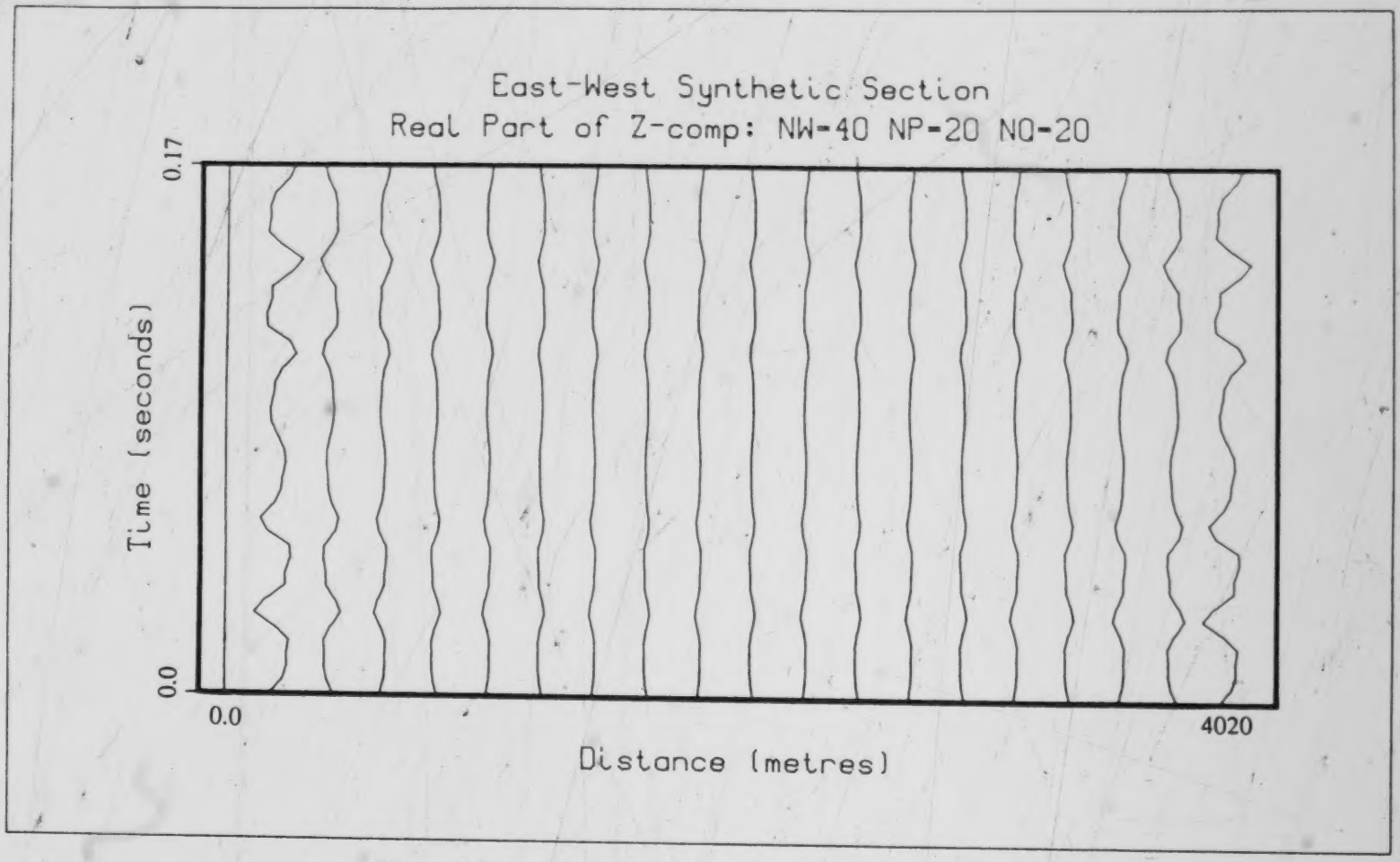

Figure 4.8 Real Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

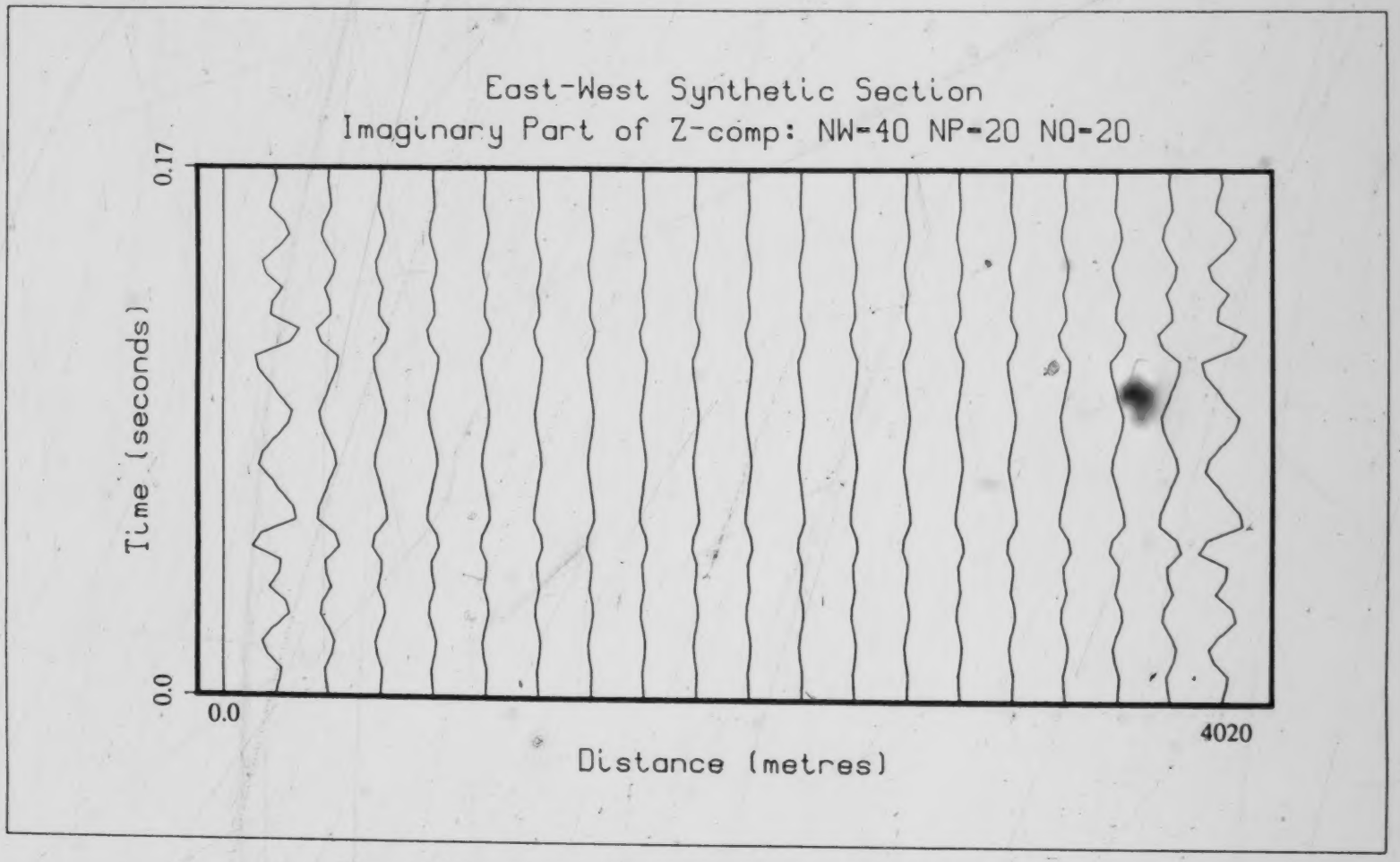

Figure 4.9 Imaginary Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

Using the modified source described above with a = b = 10, I produced synthetic seismograms. The sections that correspond to the ones shown in figures 4.1 and 4.2 are shown in figures 4.6 and 4.7. The same sections for a = b = 0.1 are shown in figures 4.8 and 4.9. Figures 4.6 and 4.8 and figures 4.7 and 4.9 are seen to be virtually identical when overlain. Again, amplitudes are not significantly changed, and other differences do not appear significant.

Another Source Formulation

A possible problem with the above source formulation is that the derivative of the delta function defined by (4.7) does not exist at 0. This can be avoided by an alternate description of the source:

$$\delta(x) = A\, e^{-x^2/a^2} \tag{4.15}$$

where A and a are constants > 0. Given (4.15), I can write

$$\int_{-\infty}^{\infty} \delta(x)\, dx = 2 \int_{0}^{\infty} A\, e^{-x^2/a^2} dx$$

$$= 2Aa^2 \tag{4.16}$$

[Gradstein and Rizhik, 1971]. And, if now the constants are chosen such that $A = 1/(2a^2)$, then (4.9) is still satisfied. It follows from (4.15) that the x-derivative of the delta function is

$$\frac{d}{dx}\delta(x) = -2\,\frac{A}{a^2}\, x\, e^{-x^2/a^2} \tag{4.17}$$

I now specify my source as

$$g_i(x,t) = M_{i1}\,\frac{2A}{a^2}\, x\, e^{-x^2/a^2}\, B\, e^{-y^2/b^2}\, C\, e^{-t^2/c^2}\, \delta(z-z_0)$$

$$+ M_{i2}\, A\, e^{-x^2/a^2}\,\frac{2B}{b^2}\, y\, e^{-y^2/b^2}\, C\, e^{-t^2/c^2}\, \delta(z-z_0)$$

$$- M_{i3}\, A e^{-x^2/a^2} B e^{-y^2/b^2} C e^{-t^2/c^2} \frac{\partial}{\partial z} \delta(z-z_0) \qquad (4.18)$$

where $A=1/(2a^2)$, $B=1/(2b^2)$, and $C=1/(2c^2)$. The Fourier transform of the first term in (4.18) is given by

$$\iiint_{-\infty}^{\infty} \left[M_{i1} \frac{2A}{a^2} x e^{-x^2/a^2} B e^{-y^2/b^2} C e^{-t^2/c^2} \delta(z-z_0) \right] e^{i(px + qy - \omega t)} dt\, dy\, dx$$

$$= M_{i1} \frac{2A^2 BCD}{a^2} \delta(z-z_0) \iint_{-\infty}^{\infty} x e^{-x^2/a^2} e^{-y^2/b^2} e^{i(px+qy)} \left[\int_{-\infty}^{\infty} e^{-t^2/c^2} \left[\cos(\omega t) - i\sin(\omega t) \right] dt \right] dx\, dy$$

$$= M_{i1} \frac{2A^2 BCD}{a^2} \delta(z-z_0) \left[\sqrt{\pi}\, c e^{-\omega^2 c^2/4} \right] \iint_{-\infty}^{\infty} x e^{-x^2/a^2} e^{-y^2/b^2} e^{i(px+qy)}\, dx\, dy$$

$$= M_{i1} \frac{2A^2 BCD}{a^2} \delta(z-z_0) \left[\sqrt{\pi}\, c e^{-\omega^2 c^2/4} \right] \left[\sqrt{\pi}\, b e^{-q^2 b^2/4} \right] \int_{-\infty}^{\infty} x e^{-x^2/a^2} e^{ipx}\, dx$$

$$= iM_{i1} A^2 BCDabcp\, \pi^{3/2} e^{-1/4(a^2p^2 + b^2q^2 + c^2\omega^2)} \delta(z-z_0) \qquad (4.19)$$

[Gradstein and Rizhik, 1971, p 494, 509]. The transforms of the second and third terms in (4.18) are found in a similiar ways, and thus g_i^* in (2.31) is now given by

$$g_i^* = iM_{i1} A^2 BCDabcp\pi^{3/2} e^{-1/4(a^2p^2 + b^2q^2 + c^2\omega^2)} + iM_{i2} AB^2 CDabcq\pi^{3/2} e^{-1/4(a^2p^2 + b^2q^2 + c^2\omega^2)} \qquad (4.20)$$

and g_i^{**} in (2.31) is given by

$$g_i^{**} = -M_{i3} ABCDabc\pi^{3/2} e^{-1/4(a^2p^2 + b^2q^2 + c^2\omega^2)} \qquad (4.21)$$

Using the modified source described above with a = b = c = 1, I again produced synthetic seismograms. The sections that correspond to the ones shown in figures 4.1 and 4.2 are shown in figures 4.10 and 4.11. Maximum amplitudes are of the order of 10^{-2}, which is now within reasonable agreement with the amplitudes I predicted at the beginning of this chapter. Also, only the trace directly above the source (x = y = 0) is zero. This is to be expected for a horizontal shearing source. Other differences do not appear significant.

To demonstrate any effect the choice of earth model has on the results, the recalculated the sections corresponding to figures 4.10 and 4.11 using a earth model closer to that representing GLISP. Model 2 consisted of a 225 metre thick layer (α = 2000 m/s, β = 1150 m/s, ρ = 2100 gm/cm^3), with the source at 200 metres depth, over a half-space with α = 2600 m/s, β = 1800 m/s, and ρ = 2400 gm/cm^3. The resulting sections are shown in figures 4.12 and 4.13 and show no important deviations from those in figures 4.10 and 4.11.

Enforcing a Real f

Of all the quantities in (2.36), only **E** does not have the symmetry necessary to transform to a real quantity. After examining (2.17), this implies that the wave that has been specified as propagating down out the stack of layers is complex. Obviously we must somehow constrain this wave to have a real amplitude in the space-time domain.

I have a vector **f**

$$\mathbf{f}(x,y,z,t) = \int\int\int_{-\infty}^{\infty} \mathbf{E}\Lambda\mathbf{w}\, e^{-i(px + qy - \omega t)}\, dp\, dq\, d\omega \tag{4.22}$$

that satisfies the basic equations (2.1) and (2.2). It follows that the real part of **f** also satisfies (2.1) and (2.2). And therefore I can write

$$\mathrm{Re}\,\mathbf{f}(x,y,z,t) = \frac{1}{2}\left[\mathbf{f}(x,y,z,t) + \mathbf{f}^*(x,y,z,t)\right]$$

$$= \frac{1}{2}\int\int\int_{-\infty}^{\infty} \left[\mathbf{E}\Lambda\mathbf{w}\, e^{-i(px + qy - \omega t)} + \mathbf{E}^*\Lambda^*\mathbf{w}^* e^{i(px + qy - \omega t)}\right] dp\, dq\, d\omega \tag{4.23}$$

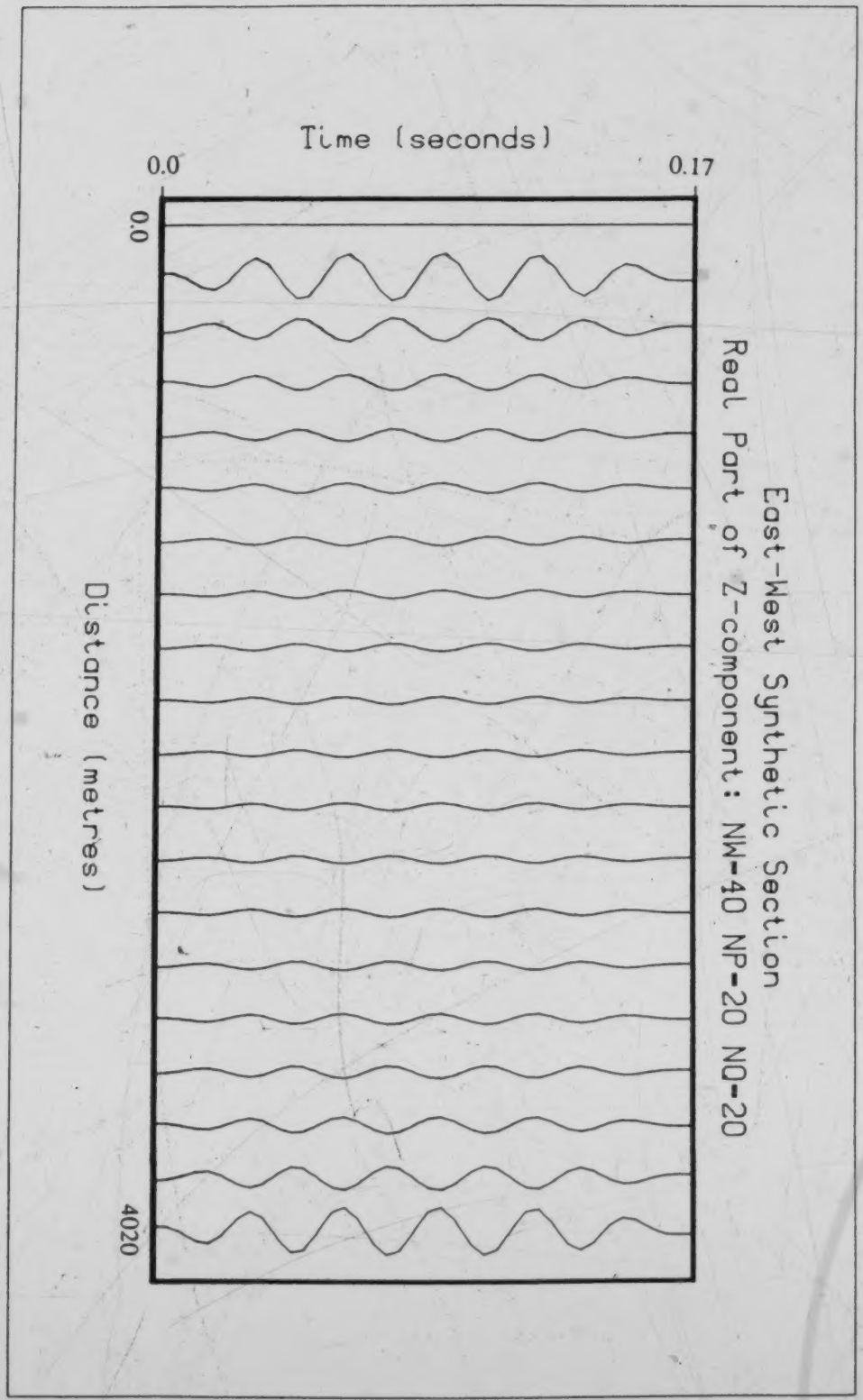

Figure 4.10 Real Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

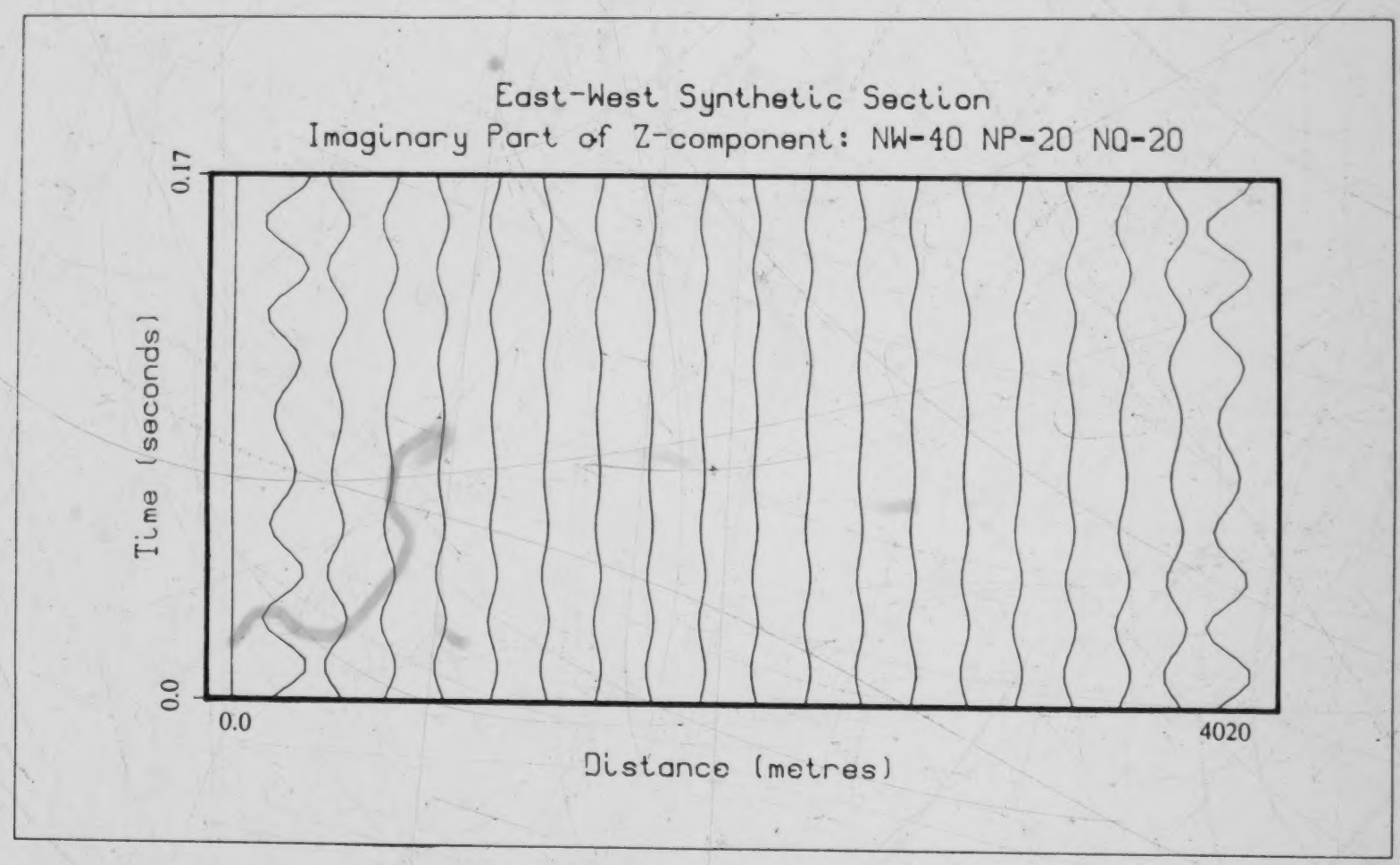

Figure 4.11 Imaginary Part of results of an East-West section for Model 1
(NW, NP, NQ = number of ω's, p's, and q's summed)

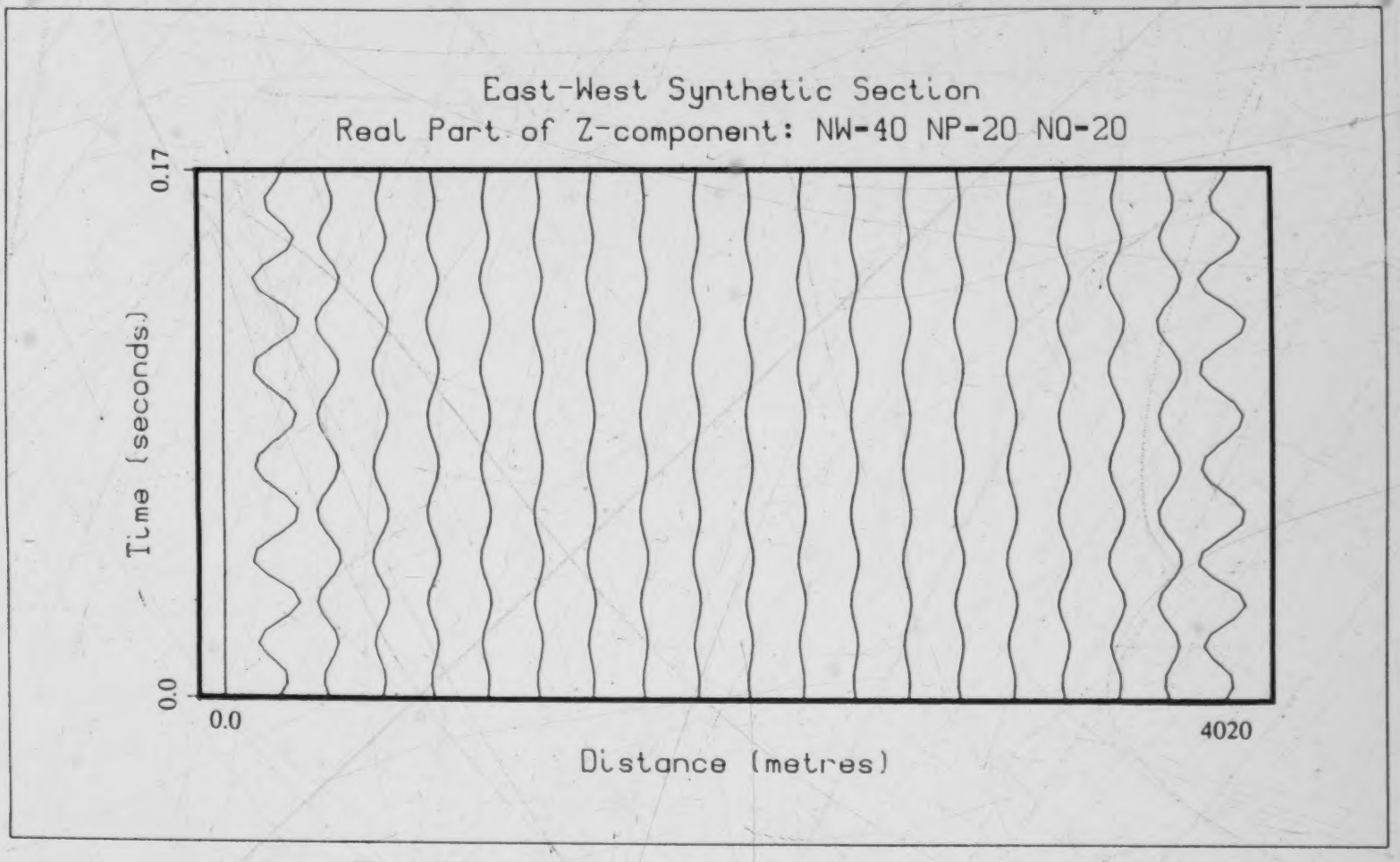

Figure 4.12 Real Part of results of an East-West section for Model 2
(NW, NP, NQ = number of ω's, p's, and q's summed)

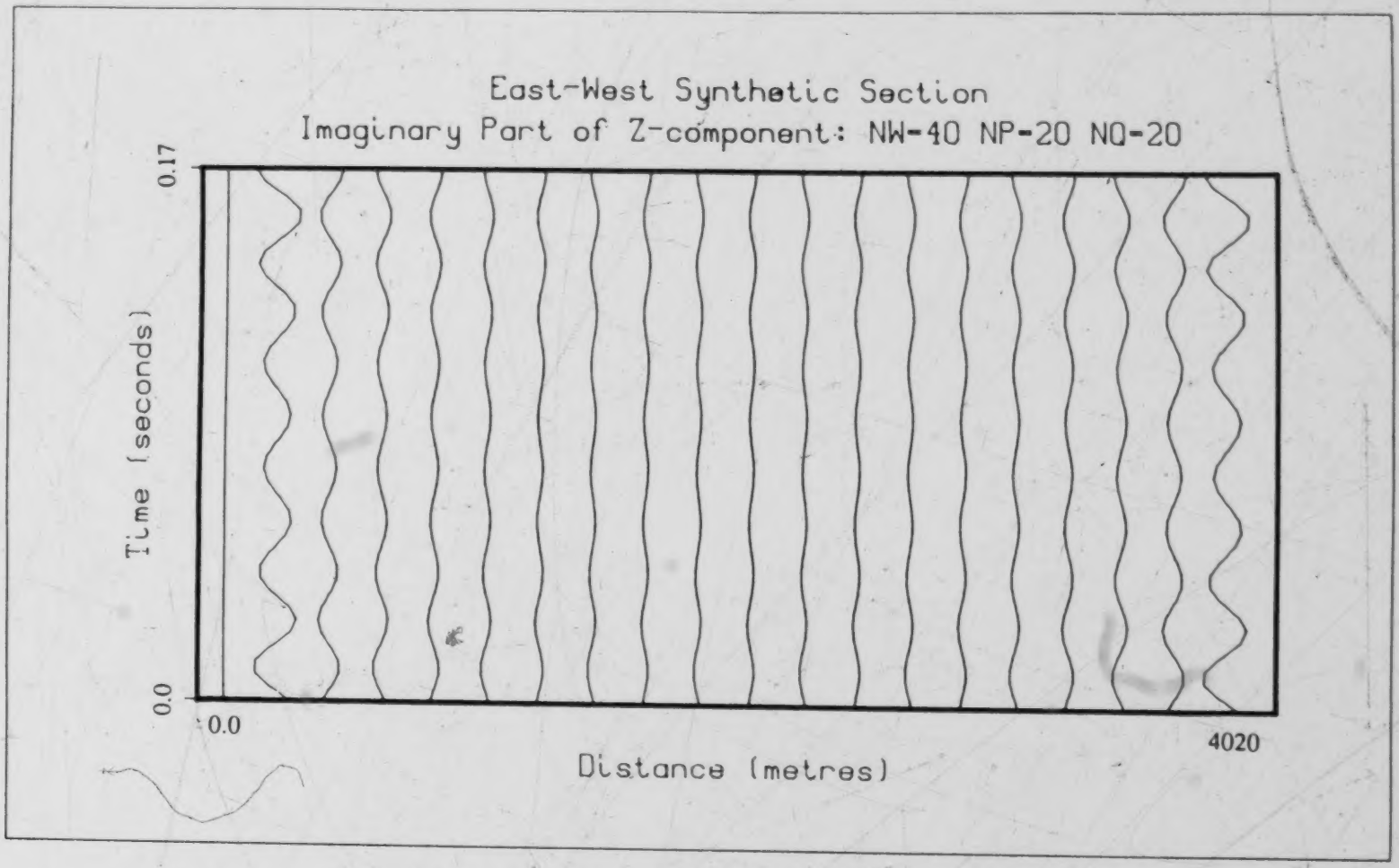

Figure 4.13 Imaginary Part of results of an East-West section for Model 2
(NW, NP, NQ = number of ω's, p's, and q's summed)

and thus

$$\mathbf{h}(p,q,z,\omega) = \int\int\int_{-\infty}^{\infty} \mathrm{Re}\left[\mathbf{f}(x,y,z,t)\right] e^{i(p'x + q'y - \omega't)} dx\, dy\, dt$$

$$= \frac{1}{2} \int\int\int_{-\infty}^{\infty} \left[\int\int\int_{-\infty}^{\infty} \mathbf{E}\Lambda \mathbf{w} e^{i[(p'-p)x+(q'-q)y-(\omega'-\omega)t)]} + \mathbf{E}^*\Lambda^*\mathbf{w}^* e^{i[(p'+p)x+(q'+q)y-(\omega'+\omega)t)]} dp dq d\omega \right] dx dy dt$$

(4.24)

but recognizing that

$$\int_{-\infty}^{\infty} e^{ipx} dx = \delta(p) \tag{4.25}$$

*and letting $w^*(-p,-q,-\omega) = w(p,q,\omega)$, I can write

$$h(p,q,0,\omega) = \left[\mathbf{E}(p,q,\omega) + \mathbf{E}^*(-p,-q,-\omega)\right] \mathbf{w} \tag{4.26}$$

Equation (4.26) shows that replacing $\mathbf{E}(p,q,\omega)$ by $\mathbf{E}(p,q,\omega)+\mathbf{E}^*(-p,-q,\omega)$ ($\mathbf{E}$ is symmetric with respect to ω) will produce a displacement-stress vector $\mathbf{h}(x,y,0,t)$ that is real and satisfies equations (2.1) and (2.2).

I replaced the eigenvector matrix by the above expression, and generated synthetics using the source given by (4.21) and (4.22). The expected symmetry of $u(p,q,\omega)$ is now present, and thus the amount of computations prior to Fourier transforming can be reduced by a factor of four using symmetry relations as described in chapter 3. A typical section is given in figure 4.14. The amplitudes are real and of the order of 10^{-2} m/s, with the trace at $x = y = 0$ a null trace. The traces are beginning to look more impulse-like and less oscillatory, but there is significant energy before the first arrival is expected.

To demonstrate the effect of the number frequencies and wave numbers summed over, figure 4.14 was recomputed for twice as many ω's, p's, and q's. The range of frequencies and wavenumbers was the same, so the sampling intervals were halved. The result is shown in figure 4.15. This section contains wavetrains that have at least the appearance of seismic arrivals. For the second trace (x = 200 m), the direct P- and S- wave

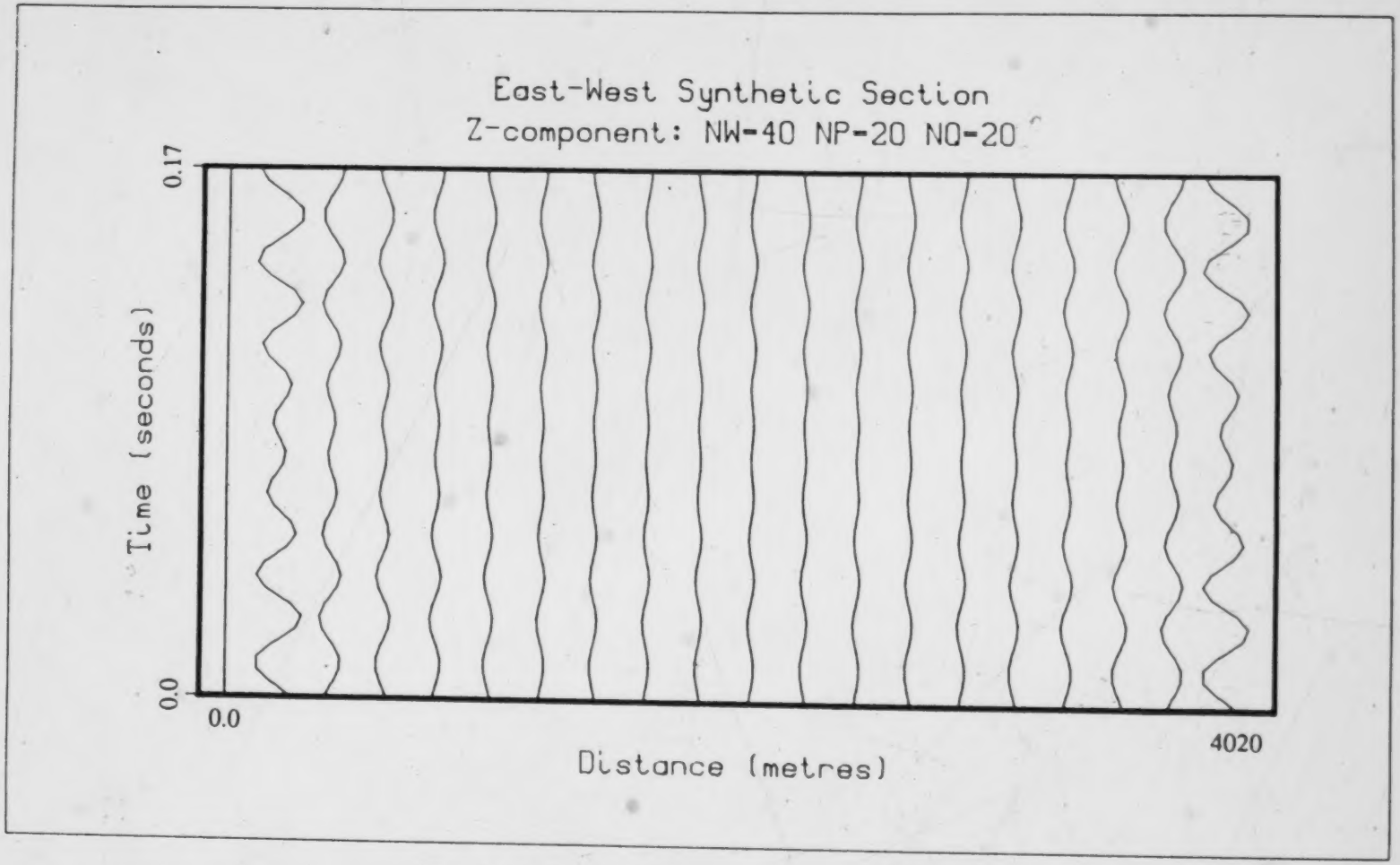

Figure 4.14 Results of an East-West section for Model 2
(NW, NP, NQ = number of ω's, p's, and q's summed)

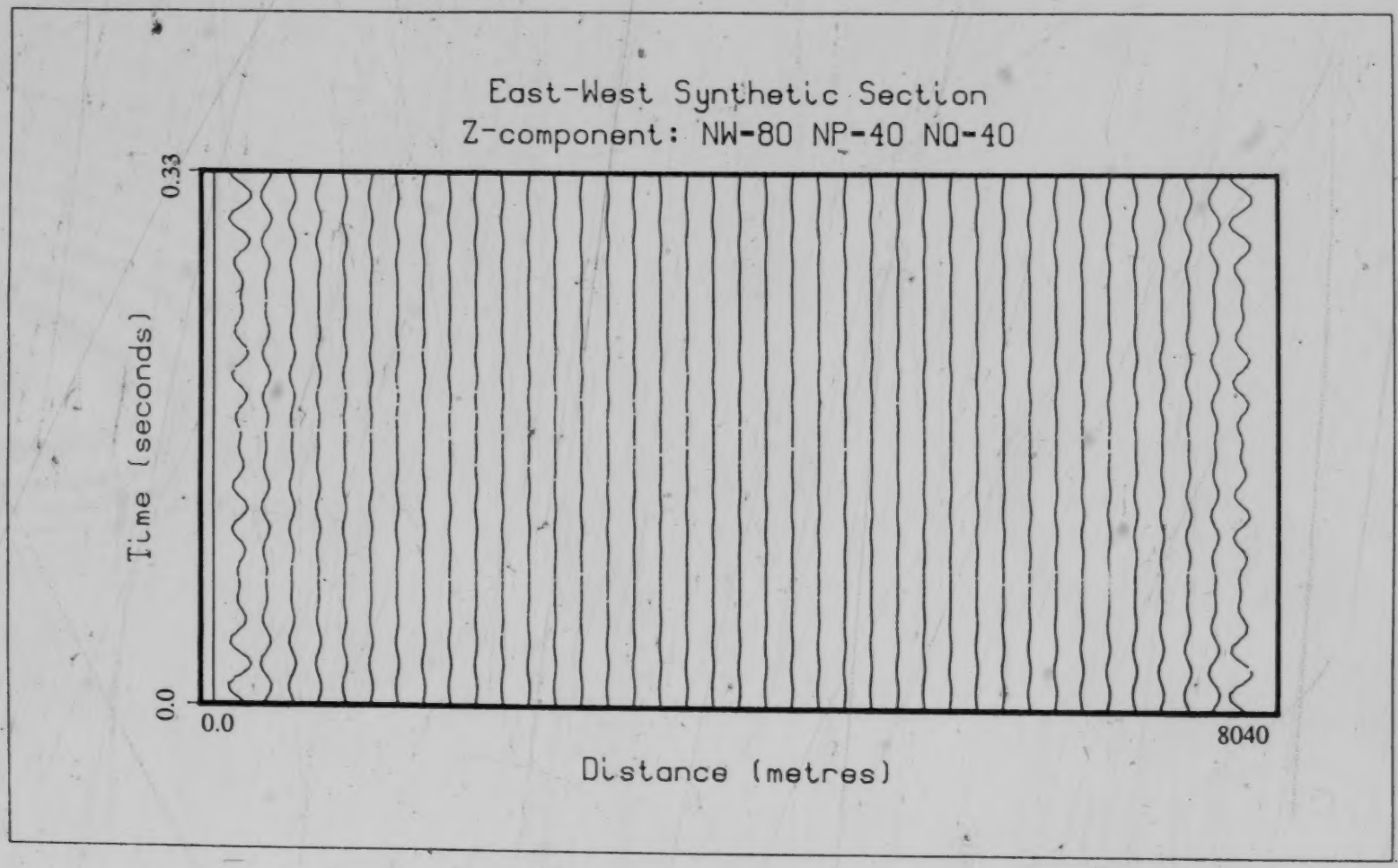

Figure 4.15 Results of an East-West section for Model 2
(NW, NP, NQ = number of ω's, p's, and q's summed)

arrivals are expected at 0.14 and 0.25 seconds respectively. An arrival is seen at 0.25 sec, but not at 0.14 sec, but a strong compressional arrival would not be expected from a shear source. The impulses at 0.25 and ~ 0.2 sec can be seen on traces 3 and 4 as well, and treating these as arrivals, and apparent velocity of > 12000 m/s is calculated for each one, and thus an interpretation of a first arrival or refraction is impossible.

An obvious thing to try would be to increase the range of the frequencies and wavenumbers summed over. This causes difficulty however, because for p's and q's even slightly larger than the values I have used to model GLISP, singularities are encountered. These singularities correspond to the existence of surface waves. As discussed previously, these singularities make inverse Fourier transforming very difficult, and as a consequence, the inverse transforming would have to be done in a complex plane with great care taken to avoid all poles and branch cuts.

Sampling in the Fourier Domain

I have an analytic expression for u(p,q,f). I define the amplitude of u to be zero for f < 30 Hz or > 90 Hz. I then sample the spectrum at frequencies separated by Δf, which is equivalent to multiplying by an infinite Dirac comb with an impulse separation of Δf, and then I inverse Fourier transform. This is equivalent to convolving u(t) with a Dirac comb with an impulse separation of $1/\Delta f$. This causes the u(t) to repeat in time every $1/\Delta f$ seconds [Kanasewich, 1981]. If the length of u(t) is longer than $1/\Delta f$, then multiples of u(t) will be folded back upon u(t) in the range 0 to $1/\Delta f$.

For figure 4.15, Δf = 240 Hz/80 = 3 Hz and thus u(t) repeats every 0.33 sec. The direct shear wave is not expected until 0.25 sec so the trace duration is obviously longer than 0.25 sec, and clearly I have a problem. The first arrival is expected to be the direct P-wave, which arrives at 0.14 sec, and energy arriving before this can now be attributed to folding of multiple copies of u(t) back upon itself it the primary range of t. Interference of this folded energy is probably disguising the true P arrival.

The sampling in p is $\Delta p = P/NP$, where P is the total range of p's being summed, and NP is the number of p's being summed. In an analogy to sampling in time and the Nyquist frequency, the Nyquist distance is given by $1/(2\Delta p) = NP/(2P)$. Since $X_{max} = NP\Delta x = NP/P$, the Nyquist distance is at the midpoint of the distance axis in figure 4.15. Again in an analogy to the time-frequency situation, just as frequencies above the Nyquist frequency are interpreted as being negative frequencies, I can interpret distances above the Nyquist distance as being negative distances, and thus I can fold the second half of figure 4.15 into the negative distance axis (see figure 4.16).

The very same argument can be applied to time axis. The sampling interval in frequency is $\Delta f = F/NF$, and thus the Nyquist time is given by $1/(2\Delta f) = NF/(2F)$. The

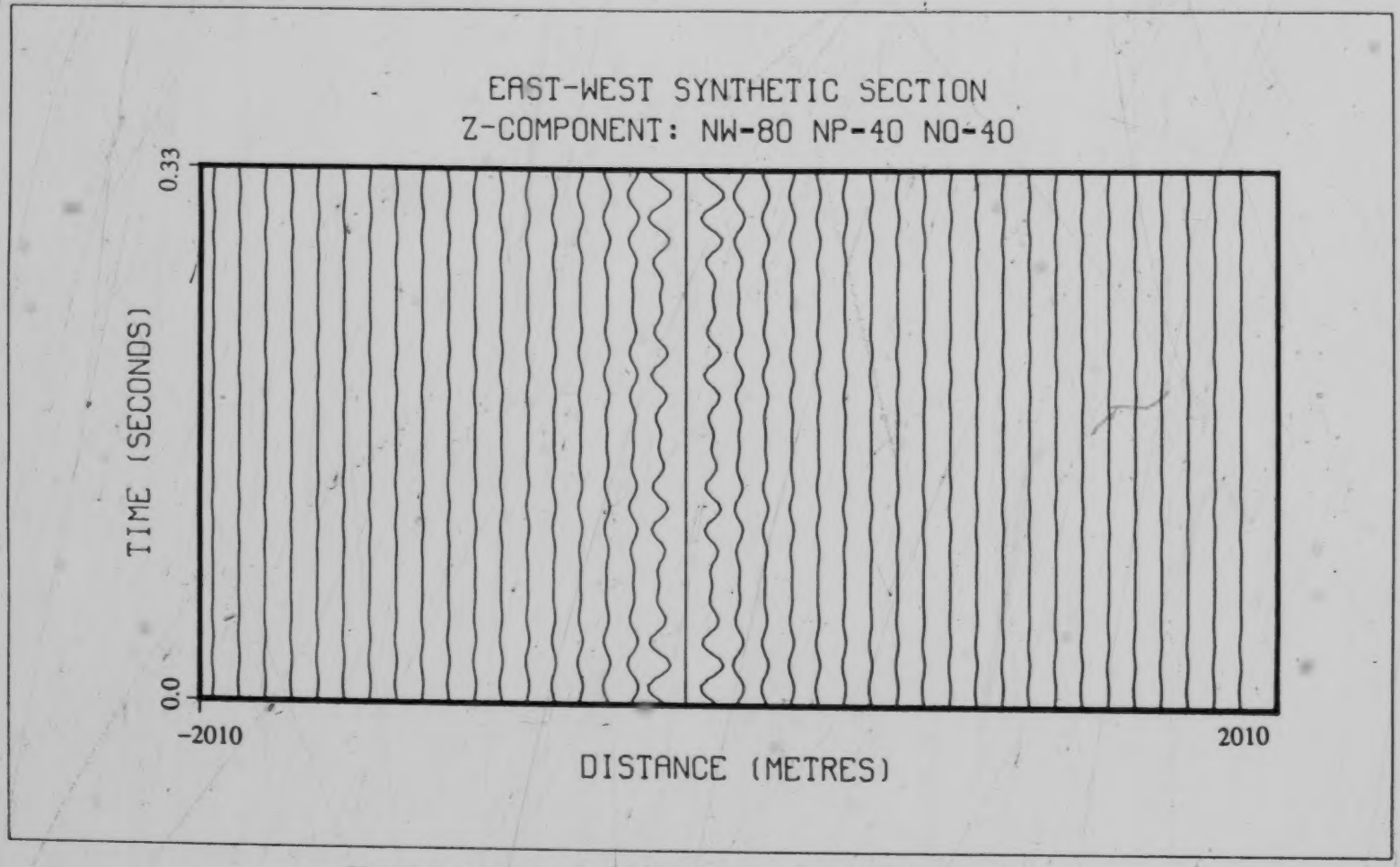

Figure 4.16 Results of an East-West section for Model 2. Traces beyond the Nyquist distance have been folded into negative distances (NW, NP, NQ = number of w's, p's, and q's summed)

total time along the time axis is $1/\Delta f$, and thus the Nyquist time is half way along the time axis, and times after this can be mapped into negative times. Since my source is at time zero, no energy can arrive at the surface before this time, and any energy existing in the trace before zero time is due to multiples of u(t) being folded upon itself.

To avoid the problem of folding multiples of u(t) back onto itself, the sampling interval in frequency (Δf) must be chosen small enough so that the period at which u(t) repeats ($1/\Delta f$) is longer than the time period from t = 0.0 to end of non-zero values of u(t). The calculation of u for 40 ω's, and 20 p's and q's that was necessary to produce the results in figure 4.15 took ~ 6 minutes of CPU time on the Amdahl mainframe at the University of Alberta. To do the calculation properly for a reasonable bandwidth of frequencies and wavenumbers, a very large amount of computing time would be required.

Chapter 5

Conclusions

The analytic expression for the full six by six seismological propagator matrix derived here is original work. Computer code for the calculation of this propagator matrix has been written and is given in Appendix 2. This is code is exact, but is inefficient and consequently is computationally expensive. It is expected that the effort required for numerical computation could be eased by algebraic simplification of the expression for the propagator matrix (this is a difficult task on REDUCE2, but may be easier on other algebraic manipulators) and by the implementation of a interpolation scheme that would interpolate between calculated values of the propagator and thus reduce the number of times that the propagator has to be calculated explicitly.

I specified a point source in terms of the seismic moment tensor times the derivative of the Dirac delta function. This allows for an arbitrary orientation of any source mechanism whether it be shearing or explosive, and once integrated in space and time it allows for sources of finite size. The source must actually be specified in terms of approximations of Dirac delta functions rather than true delta functions as the derivative of the Dirac delta function is not square integrable and thus does not have a Fourier transform. I suggest that an approximation in terms of Gaussian functions is appropriate because the condition of unit impulse can easily be conserved, constants describing the source can be chosen considering physical significance, and the source can be made arbitrarily close to a delta function constrained only by the numerical limits of the digital computer used. The fault plane area for microseismic sources is thought to be of the order of 1 m^2, the constants of the Gaussians can be chosen so that the specified source has dimensions of order of 1 m^2. Having a source description with finite area does not contradict a point source description, because as observed seismic radiation has wavelengths of the order of 100 m, and thus a source dimension of 1 m is still effectively a point source.

I encountered numerical difficulties in computing the inverse Fourier transform. These difficulties are a consequence of the periodic nature of the untransformed result. If I sample u(f) with a sampling interval Δf, the untransformed u(t) will be repeated every $1/\Delta f$. If the signal u(t) has non-zero amplitudes for times greater than $1/\Delta f$, then this energy will show up in the primary range of times from 0 to $1/\Delta f$. The true signal u(t) is of course analogue and transient. The induced periodicity is purely an artifact of sampling u(f), and thus if energy is being folded into the primary range of times, the signal u(t) will show non-physical effects that are due solely to sampling in the frequency domain.

The problem is that given the frequency spectrum of a signal u(f), there is no way to determine a priori the length of u(t) (when I say the length of u(t), I mean the time T for

which u is zero for all times $t \geq T$). Therefore, the appropriate choice of Δf can only be verified by comparison of different u(t)'s produced with different Δf's. Once Δf has been chosen small enough, further reductions in its size will not effect u(t).

Given sufficient computer resources, the next step in process of modelling GLISP seismograms by the method I have shown would be to choose Δf so that $1/\Delta f$ is greater than the length of the expected seismic wavetrain. This length will depend on the seismic model and types of waves present. The observed wavetrains during steaming at GLISP were of the order of 1 sec, and thus a Δf of 1 Hz might be required (the smallest Δf I used was 3 Hz, and this was not decreased due to limitations in computer time). Once this is done, one should be able to make reasonable comparisons between the calculated seismograms and the ones observed at GLISP.

Following this, one should modify the location and description of the source until the correlation between observed and calculated seismograms is maximized.

To improve the resemblance of the two sets of seismograms, spatial aliasing would have to be considered in the calculated seismograms. It is thought probable that significant spatial aliasing occurred in the seismometer array during the monitoring at GLISP. For this reason, waves with wavelengths and wavenumbers outside the ranges I considered are contributing to observed ground displacements. The amplitudes of these waves would have to be calculated, and the calculation of the synthetic seismograms would have to be modified to consider the aliasing of these waves. Again, modelling parameters could be changed and the synthetic seismograms recomputed. The effects on the computed seismograms, when compared to the observed seismograms, will dictate whether the changed parameters do a better or worse job of modelling GLISP.

The effect of widening the bandwidths of frequency and wavenumber is that types of waves other than the ones I have considered are encountered. This complicates the problem, because now singularities are encountered during the calculation of the surface displacement. To avoid these singularities, one must compute the Fourier integrals in a complex plane in the presence of poles and branch cuts. Complex contour paths must be followed to avoid these complicating points and lines. It is presently unclear how the use of a Fast Fourier Transform can be preserved in this situation.

The lengthening of bandwidths is only necessitated by the presence of spatially aliased waves. Intelligent comparisons can probably be made between observed and calculated seismograms without considering aliased waves in the calculated seismograms.

I have solved the problem for an arbitrary number of layers, making no assumptions about density or seismic velocity within the layers. Also, the source may be buried at any point within the stack of layers. However, to simplify the problem, I have assumed the layered earth model to be isotropic, and a natural follow-up of this work would be to reformulate the equations in order to consider an earth consisting of a stack of layered anisotropic material, a more realistic model particularly for the shallow sediments found in

northern Alberta

Although the mathematical complexity that was necessary for this theoretical work was greater than initially anticipated, the objective of this thesis is unchanged and is the modelling of observations made at GLISP. The desire to model the observations made at GLISP is motivated by a desire to gain a better understanding of the mechanism, geometry and location of the propagating steam front. As it stands now, little is understood about the direction and the mechanism by which the induced fracture propagates, as well as the overall shape of the induced fracture. A solid understanding of a realistic method for modelling the observations made at GLISP should, at least, provide insight into the fracturing mechanism and the location of the steam front tip.

Finally, I comment that the procedure used for the calculation of the propagator can easily be followed for a generalized situation. That is, given a layered media of any linear form that can be described by a constitutive equation of the form of (2.1) the propagator matrix can be derived in the manner I have demonstrated. For example, a two phase saturated oil sand where one phase is the sand matrix that obeys the elastic equations I have specified, and a second phase consisting of bitumen that is subject to visco-elastic behavior. This situation can be considered by adding D'Arcy's equation, a linear constitutive equation, and the continuity equation for fluid flow to the original six equations. The continuity equation is needed to eliminate the fluid flow velocity in terms of fluid content and fluid pressure.

Bibliography

Aki, K., and Richards, P.G., 1980, Quantitative seismology, W.H. Freeman and Company.

Bailey, N.J.L., Jobson, A.M., and Rogers, M.A., 1973, Bacterial degradation of crude oil: Comparison of field and experimental data, Chemical Geology, vol. 11, p. 203-221.

Carrigy, M.A., 1959, Geology of the McMurray Formation Pt. III, General geology of the McMuray area: Research Council of Alberta, Memoir 1, 130p.

Churchill, R. V. and Brown, J., W., 1984, Complex variables and applications, McGraw-Hill Book Company, 399 pp.

Deif, A. S., 1982, Advanced matrix theory for scientists and engineers, Abacus Press, 241 pp.

Demaison, G.J., 1977, Tar sands and supergiant oil fields: The oil sands of Canada-Venezuela, CIM special volume 17, p. 9-16.

Dusseault, M. B., 1977, Stress state and hydrolic fraturing in the Athabasca Oil Sands: The oil sands of Canada-Venezuela, CIM special volume 17, p. 27-35.

Dusseault, M.B. and Rothenburg, L., 1988, Shear dilatency and permeability enhancement in oil sands, paper #32, preprints vol. 1, Fourth UNITAR/UNDP conference on heavy crude and tar sands, August, 1988, Emonton, Alberta.

Farouq Ali, S. M., 1974, Current status of steam injection as a Heavy oil recovery method, The Journal of Canadian Petroleum Technology (Jan. 1974), p. 54-68.

Flock, D.L. and Lee, J., 1977, An experimental investigation of steam displacement of a medium gravity crude oil: The oil sands of Canada-Venezuela, CIM special volume 17, p. 386-394.

Goertzel, G. and Tralli, N., 1960, Some mathematical methods of physics, McGraw-Hill Book Company, 300 pp.

Gradstein, I.S. and Rizhik, J.M., 1971, Tables of Integrals, Sums, Series and Products, Nauka 1108 pp. *in Russian*.

Harris, M.C. and Sobkowicz, J.C., 1977, Engineering behavior of oil sand: The oil sands of Canada-Venezuela, CIM special volume 17, p. 270-281.

Hearn, A. C., 1982, REDUCE2 User's Manual Second Edition, University of Utah, Salt Lake City, Utah 84112, USA.

Hodgman, C.D., Samuel, S.M., Weast, R.C. (eds), 1959, Mathematical tables from the handbook of chemistry and physics, The Chemical Rubber Publishing Company, 447 pp.

James, D.P. and Oliver, T.A., 1977, The Sedimentology of the McMurray Formation, East Athabasca: The oil sands of Canada-Venezuela, CIM special volume 17, p. 17-26.

Kanasewich, E.R., 1981, Time sequence analysis in geophysics, The University of Alberta Press, 480 pp.

Macrides, Costas G., 1987, Seismic tomography in oil sands for monitoring thermal recovery processes, PhD thesis, The University of Alberta, Edmonton, Alberta.

Pullin, N., Mattews, L., and Hirsche, K., 1987, Techniques applied to obtain very high resolution 3-D imaging at an Athabasca tar sands thermal pilot, The Leading Edge, vol. 6, No. 9, p. 10-15.

Selly, R. C., 1985, Elements of petroleum geology, W.H. Freeman and Company, 449 pp.

Sneddon, I. N., 1951, Fourier Transforms, McGraw Hill Book Company, 542 pp.

Appendix 1

Algebraic Computations with REDUCE2

For REDUCE2, I formulated the three relations (i = 1, 2, 3) represented by equation (2.1) as:

$$0 = \frac{d\sigma_{11}(z)}{dx} + \frac{d\sigma_{12}(z)}{dy} + \frac{d\sigma_{13}(z)}{dz} + \rho\omega^2 u(z) \qquad \text{(A1.1)}$$

$$0 = \frac{d\sigma_{12}(z)}{dx} + \frac{d\sigma_{22}(z)}{dy} + \frac{d\sigma_{23}(z)}{dz} + \rho\omega^2 v(z) \qquad \text{(A1.2)}$$

$$0 = \frac{d\sigma_{13}(z)}{dx} + \frac{d\sigma_{23}(z)}{dy} + \frac{d\sigma_{33}(z)}{dz} + \rho\omega^2 w(z) \qquad \text{(A1.3)}$$

where the property $\sigma_{ij} = \sigma_{ji}$ has been used. I have neglected the source contributions g_i, since they have coefficients of 1 and the source vector is easily found via (2.6). For i = 1, 2, 3 , and j = 1, equation (2.2) was represented as:

$$0 = \sigma_{13}(z) - \mu\left[\frac{du(z)}{dz} + \frac{dw(z)}{dx}\right] \qquad \text{(A1.4)}$$

$$0 = \sigma_{23}(z) - \mu\left[\frac{dv(z)}{dz} + \frac{dw(z)}{dy}\right] \qquad \text{(A1.5)}$$

$$0 = \sigma_{33}(z) - \lambda\left[\frac{du(z)}{dx} + \frac{dv(z)}{dy} + \frac{dw(z)}{dz}\right] - 2\mu\left[\frac{dw(z)}{dz}\right] \qquad \text{(A1.6)}$$

where, since I am working in the transformed domain, the specified derivatives with respect to x and y are very simple

$$\frac{d}{dx} f = \sqrt{-1}\, p\, f$$

$$\frac{d}{dy} f = \sqrt{-1}\, q\, f$$

where f is any eligible function (ie. u(z), v(z), w(z), or σ(z)). By making the substitutions (A1.7) - (A1.9):

$$\sigma_{11}(z) = \lambda\left[\frac{du(z)}{dx} + \frac{dv(z)}{dy} + \frac{dw(z)}{dz}\right] + 2\mu\frac{du(z)}{dx} \tag{A1.7}$$

$$\sigma_{12}(z) = \mu\left[\frac{du(z)}{dx} + \frac{dv(z)}{dy}\right] \tag{A1.8}$$

$$\sigma_{22}(z) = \lambda\left[\frac{du(z)}{dx} + \frac{dv(z)}{dy} + \frac{dw(z)}{dz}\right] + 2\mu\frac{dv(z)}{dy} \tag{A1.9}$$

also as defined by equation (2.1). I now have six equations whose only unknowns are the 6 components of **f**, and the first derivative, with respect to z, of the components of **f**.

Using REDUCE2, I can determine the coefficients of f(z) and df(z)/dz and thus form the following equation:

$$0 = \mathbf{A}_l\frac{d\mathbf{f}(z)}{dz} + \mathbf{A}_r\mathbf{f}(z) \tag{A1.10}$$

and so the matrix **A** I look for is:

$$\mathbf{A} = -\mathbf{A}_L\,\mathbf{A}_R. \tag{2.11}$$

where the source term is given by (2.6)

$$\mathbf{g}(z) = -\mathbf{A}_L^{-1}\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -g_1 \\ -g_2 \\ -g_3 \end{bmatrix} \tag{2.6 (again)}$$

The result is simplified by enforcing some substitutions:

$$\lambda \Rightarrow \rho(\alpha^2 - 2\beta^2) \qquad \mu \Rightarrow \rho\beta^2$$

$$p^2 + q^2 \Rightarrow k^2$$

The next step was to calculate the propagator matrix via equation (2.15). Since $X\mathbf{1} - \mathbf{A}$ can only be singular at the eigenvalues of **A**, I need only calculate the residues of $X\mathbf{1} - \mathbf{A}$

at the four distinct eigenvalues of **A**:

$$\mathbf{P}(z,z_0) = e^{(z-z_0)\mathbf{A}}$$

$$= \frac{1}{2\pi\sqrt{-1}} \oint_C \frac{e^{(z-z_0)X}}{X\cdot 1 - \mathbf{A}}\, dX$$

$$= \frac{1}{2\pi\sqrt{-1}} 2\pi\sqrt{-1} \sum_i \mathrm{Res}_i$$

$$= \left[\mathrm{Res}\left[\frac{e^{(z-z_0)\mathbf{A}}}{X-\mathbf{A}}, \lambda_1\right] + \mathrm{Res}\left[\frac{e^{(z-z_0)\mathbf{A}}}{X-\mathbf{A}}, \lambda_2\right] + \mathrm{Res}\left[\frac{e^{(z-z_0)\mathbf{A}}}{X-\mathbf{A}}, \lambda_4\right] + \mathrm{Res}\left[\frac{e^{(z-z_0)\mathbf{A}}}{X-\mathbf{A}}, \lambda_6\right]\right]$$

(A1.12)

since λ_2 and λ_3, and λ_4 and λ_5 are identical due to the degeneracy of shear waves in an isotropic medium.

Let us define a 6 x 6 matrix Ψ

$$\Psi = \frac{e^{(z-z_0)\mathbf{A}}}{X-\mathbf{A}} \tag{A1.13}$$

It is known that Ψ is singular only at the eigenvalues of **A**. It follows, that I can express each component of Ψ (Ψ_{lm}) as a ratio of some analytic complex function ϕ and a polynomial of the eigenvalues of **A**:

$$\Psi_{lm}(x) = \frac{\phi_{lm}(x)}{[x-\lambda_1][x-\lambda_2][x-\lambda_3][x-\lambda_4][x-\lambda_5][x-\lambda_6]} \tag{A1.14}$$

After making the following substitutions

$$\eta_\alpha^2 \Rightarrow \frac{\omega^2}{\alpha^2} - k^2 \qquad \eta_\beta^2 \Rightarrow \frac{\omega^2}{\beta^2} - k^2$$

I can write

$$\Psi_{lm}(x) = \frac{\phi_{lm}(x)}{\left[x-\sqrt{-1}\,\eta_\alpha\right]\left[x+\sqrt{-1}\,\eta_\alpha\right]\left[x-\sqrt{-1}\,\eta_\beta\right]^2\left[x+\sqrt{-1}\,\eta_\beta\right]^2} \qquad \text{(A1.15)}$$

The residues of a function of this form are easily calculated:

$$(\text{Res}_i)_{lm} = \text{Res}\left[\Psi_{lm}(x)\right]_{x=\sqrt{-1}\,\eta_\alpha}$$

$$= \text{Res}\left[\frac{1}{x-\sqrt{-1}\,\eta_\alpha}\,\frac{\phi_{lm}(x)}{\left[x+\sqrt{-1}\,\eta_\alpha\right]\left[x-\sqrt{-1}\,\eta_\beta\right]^2\left[x+\sqrt{-1}\,\eta_\beta\right]^2}\right]_{x=\sqrt{-1}\,\eta_\alpha}$$

$$= \left[\frac{\phi_{lm}(x)}{\left[x+\sqrt{-1}\,\eta_\alpha\right]\left[x-\sqrt{-1}\,\eta_\beta\right]^2\left[x+\sqrt{-1}\,\eta_\beta\right]^2}\right]_{x=\sqrt{-1}\,\eta_\alpha}$$

$$= \Phi_{lm}(x)\Big|_{x=\sqrt{-1}\,\eta_\alpha} \qquad \text{(A1.16)}$$

[Churchill and Brown, 1984].

After making the substitution $x = \sqrt{-1}\,\eta_\alpha + \varepsilon$, for each component of Ψ (Ψ_{lm}) I can write

$$\lim_{\varepsilon\to 0}\left[\varepsilon\,\Psi(\varepsilon)_{lm}\right] =$$

$$= \lim_{\varepsilon\to 0}\left[\frac{\varepsilon\,\phi_{lm}(\varepsilon)}{\left[(\sqrt{-1}\,\eta_\alpha+\varepsilon)-\sqrt{-1}\,\eta_\alpha\right]\left[(\sqrt{-1}\,\eta_\alpha+\varepsilon)+\sqrt{-1}\,\eta_\alpha\right]\left[(\sqrt{-1}\,\eta_\alpha+\varepsilon)-\sqrt{-1}\,\eta_\beta\right]^2\left[(\sqrt{-1}\,\eta_\alpha+\varepsilon)+\sqrt{-1}\,\eta_\beta\right]^2}\right]$$

$$= \lim_{\varepsilon \to 0} \left[\frac{\phi_{lm}(\varepsilon)}{\left[(\sqrt{-1}\,\eta_\alpha + \varepsilon) + \sqrt{-1}\,\eta_\alpha \right] \left[(\sqrt{-1}\,\eta_\alpha + \varepsilon) - \sqrt{-1}\,\eta_\beta \right]^2 \left[(\sqrt{-1}\,\eta_\alpha + \varepsilon) + \sqrt{-1}\,\eta_\beta \right]^2} \right]$$

$$= \Phi_{lm}(x) \Big|_{x=\sqrt{-1}\,\eta_\alpha}$$

$$= (\mathrm{Res}_1)_{lm} \qquad \text{(A1.17)}$$

(A1.17) demonstrates that the numerical calculation of the residue of the first simple poles can be accomplished by substituting $x = \sqrt{-1}\,\eta_\alpha + \varepsilon$ into $\varepsilon\Psi$ and then letting $\varepsilon = 0$. This is easily done with REDUCE2.

One possible method of calculating the S-wave pole residues follows a development analogous to that demonstrated for the P-wave pole. I start by first writing

$$(\mathrm{Res}_2)_{lm} = \mathrm{Res}\left[\Psi_{lm}(x) \right]_{x=\sqrt{-1}\,\eta_\beta}$$

$$= \mathrm{Res}\left[\frac{1}{\left[x - \sqrt{-1}\,\eta_\beta \right]^2} \frac{\phi_{lm}(x)}{\left[x - \sqrt{-1}\,\eta_\alpha \right] \left[x + \sqrt{-1}\,\eta_\alpha \right] \left[x + \sqrt{-1}\,\eta_\beta \right]^2} \right]_{x=\sqrt{-1}\,\eta_\beta}$$

$$= \frac{d}{dx}\left[\frac{\phi_{lm}(x)}{\left[x - \sqrt{-1}\,\eta_\alpha \right] \left[x + \sqrt{-1}\,\eta_\alpha \right] \left[x + \sqrt{-1}\,\eta_\beta \right]^2} \right]_{x=\sqrt{-1}\,\eta_\beta}$$

$$= \frac{d}{dx} \tau_{lm}(x) \Big|_{x=\sqrt{-1}\,\eta_\beta} \qquad \text{(A1.18)}$$

[Churchill and Brown, 1984]. In principle, I should be able to find the residue of this second order pole by following a procedure similar to that followed for the simple pole:

$$\lim_{\varepsilon \to 0} \frac{d}{d\varepsilon}\left[\varepsilon^2 \Psi(\varepsilon)_{lm} \right] =$$

$$= \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \left[\frac{\varepsilon^2 \phi_{lm}(\varepsilon)}{\left[(\sqrt{-1}\eta_\beta+\varepsilon)-\sqrt{-1}\eta_\alpha\right]\left[(\sqrt{-1}\eta_\beta+\varepsilon)+\sqrt{-1}\eta_\alpha\right]\left[(\sqrt{-1}\eta_\beta+\varepsilon)-\sqrt{-1}\eta_\beta\right]^2\left[(\sqrt{-1}\eta_\beta+\varepsilon)+\sqrt{-1}\eta_\beta\right]^2} \right]$$

$$= \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \left[\frac{\phi_{lm}(\varepsilon)}{\left[(\sqrt{-1}\eta_\beta+\varepsilon)-\sqrt{-1}\eta_\alpha\right]\left[(\sqrt{-1}\eta_\beta+\varepsilon)+\sqrt{-1}\eta_\alpha\right]\left[(\sqrt{-1}\eta_\beta+\varepsilon)+\sqrt{-1}\eta_\beta\right]^2} \right]$$

$$= \frac{d}{dx} \tau_{lm}(x) \Big|_{x=\sqrt{-1}\,\eta_\beta}$$

$$= (Res_2)_{lm} \qquad \text{(A1.19)}$$

Analogous to (A1.17), (A1.19) shows that the numerical calculation of the second order poles is equivalent to substituting $x = \sqrt{-1}\,\eta_\beta + \varepsilon$ into $\varepsilon^2\Psi$, taking $\frac{d}{d\varepsilon}$, and then letting $\varepsilon = 0$.

There is a problem however. REDUCE2 is infallible in the sense that it never makes mistakes in what it does, but it does not always evaluate expressions particularly intelligently. In this case REDUCE2 was unable to do the obvious cancellation in the expression $\varepsilon^2\Psi$, and produced an intermediate result that had a zero denominator. I invested a certain amount of effort, using the tools available to the REDUCE2 programmer (ie. flags such as KORDER, FACTOR, GCD, MCD, etc.) but we were unable to force the cancellations necessary to allow the evaluation of $\varepsilon^2\Psi$.

After several attempts to overcome this problem, I finally arrived at the following procedure which allows the evaluation of the second order S-wave poles. Firstly, I make the substitution $x = \sqrt{-1}\,\eta_\beta + \varepsilon$ and express $\varepsilon^2\Psi$ in terms of a ratio of polynomials of ε. After setting $z_0 = 0$ in (A1.13), it follows that

$$(Res_2)_{lm} = \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \left[\varepsilon^2 \Psi_{lm}(\varepsilon) \right]$$

$$= \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \left[\frac{\varepsilon^2 N_{lm}(\varepsilon)}{D_{lm}(\varepsilon)} e^{(\sqrt{-1}\,\eta_\beta + \varepsilon)z} \right]$$

$$= \lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} \left[\frac{\left[N'_{lm}(\varepsilon)\, e^{\varepsilon z} + N_{lm}(\varepsilon)\, z\, e^{\varepsilon z} \right] \frac{D_{lm}(\varepsilon)}{\varepsilon^2} - N_{lm}(\varepsilon)\, e^{\varepsilon z} \left[\frac{D_{lm}(\varepsilon)}{\varepsilon^2} \right]'}{\left[\frac{D_{lm}(\varepsilon)}{\varepsilon^2} \right]^2} e^{\sqrt{-1}\,\eta_\beta z} \right]$$

(A1.19)

where primes indicate derivatives with respect to ε.

After examining all the components of Ψ, it was determined that N was of order 0 in ε, and D was of the order of ε^2; that is:

$$N(\varepsilon) = N_0 + N_1 \varepsilon + N_2 \varepsilon^2 + \ldots$$

$$D(\varepsilon) = D_2 \varepsilon^2 + D_3 \varepsilon^3 + D_4 \varepsilon^4 + \ldots$$

and thus

$(Res_2)_{lm} =$

$$= \lim_{\varepsilon \to 0} \left[\frac{\left\{ \left[\left(N_1 + 2N_2\varepsilon + \ldots \right) + \left(N_0 + N_1\varepsilon + \ldots \right) z \right] \left[D_2 + D_3\varepsilon + \ldots \right] \right.}{\left[D_2 + D_3\varepsilon + \ldots \right]^2} \right.$$

$$\left. \frac{- \left[N_0 + N_1\varepsilon + \ldots \right] \left[D_3 + 2D_4\varepsilon + \ldots \right] \Big\} e^{\varepsilon z}}{} \right] e^{\sqrt{-1}\,\eta_\beta z}$$

$$= \frac{\left[N_1 + N_0 z\right] D_2 - N_0 D_3}{D_2^2} e^{\sqrt{-1}\,\eta_\beta z}$$

$$= \left[\frac{N_1}{D_2} + \frac{N_0}{D_2}\left[z - \frac{D_3}{D_2}\right]\right] e^{\sqrt{-1}\,\eta_\beta z} \qquad \text{(A1.20)}$$

So, the calculation of the S-pole residues can be reduced to determining three coefficients of ε in the numerator and denominator of Ψ after substituting $x = \lambda_2 + \varepsilon$ or $x = \lambda_4 + \varepsilon$, and then evaluating (A1.20). This was a simple procedure in REDUCE2.

REDUCE2 Code for Calculation of the Propagator Matrix

```
operator u,t;
on nero;off nat;
let lam=alp**2*rho-2*mu, mu=bet**2*rho;
matrix s(3,3),gg(3,1),eqns(6,1),r(3,1),b(6,6),c(6,6),a(6,6);
r:=mat((x),(y),(z));
Comment specify derivatives wrt x and y
  for all l,x,y,z let
    df(u(l,x,y,z),x)=i*p*u(l,x,y,z),
    df(u(l,x,y,z),y)=i*q*u(l,x,y,z),
    df(t(l,x,y,z),x)=i*p*t(l,x,y,z),
    df(t(l,x,y,z),y)=i*q*t(l,x,y,z);
Comment specify equations A1.7 - A1.9
for k:=1:2 do begin
  s(k,k):=lam*(for n:=1:3 sum df(u(n,x,y,z),r(n,1)));
  s(k,3):=t(k,x,y,z); s(3,k):=t(k,x,y,z);
for m:=1:2 do
  s(k,m):=s(k,m)+mu*(df(u(k,x,y,z),r(m,1))+df(u(m,x,y,z),r(k,1)))
end;
s(3,3):=t(3,x,y,z);
Comment specify equations A1.1 - A1.6
for k:=1:3 do begin
  eqns(k,1):=(for m:=1:3 sum
    df(s(k,m),r(m,1)))+rho*o**2*u(k,x,y,z);
  eqns(k+3,1):=t(k,x,y,z)-mu*(df(u(k,x,y,z),r(3,1))
                    +df(u(3,x,y,z),r(k,1))) end;
eqns(6,1):=eqns(6,1)-lam*(for n:=1:3 sum df(u(n,x,y,z),r(n,1)));
Comment find matrices A1(called C) and Ar(called B); then calc A
for l:=1:6 do
for m:=1:3 do begin
  cc1:=0;coeff(eqns(l,1),u(m,x,y,z),cc);b(l,m):=cc1;
  dd1:=0;coeff(eqns(l,1),t(m,x,y,z),dd);b(l,m+3):=dd1;
  cc1:=0;ck:=df(u(m,x,y,z),z);coeff(eqns(l,1),ck,cc);c(l,m):=cc1;
  dd1:=0;dk:=df(t(m,x,y,z),z);coeff(eqns(l,1),dk,dd);
    c(l,m+3):=dd1
end;
a:=-(1/c)*b;;
matrix w(6,6),wi(6,6);
```

```
w:=xx*(ident 6)-a;
Comment propagator matrix is contour integral of wi
out "-wi";write wi:=1/w;out t;
Comment  calculate residues
matrix prop(6,6);korder eps,eaz,ebz,o,alp,bet;
let eta**2=o**2/alp**2-(p**2+q**2),
    etb**2=o**2/bet**2-(p**2+q**2),
    p**2+q**2=k**2;
for l:=1:6 do for m:=1:6 do begin;
  write l,m;
  numer:=num(wi(l,m))$denom:=den(wi(l,m))$
Comment Residues from P-wave poles
  let xx=i*eta+eps; res1:=sub(eps=0,eps*wi(l,m)*eaz);
  let xx=-i*eta+eps; res2:=sub(eps=0,eps*wi(l,m)/eaz);
Comment Residues from S-wave poles
  let xx=i*etb+eps;n0:=n1:=d2:=d3:=0;
    coeff(numer,eps,n);coeff(denom,eps,d);
    res3:=(n1/d2+n0/d2*(dz-d3/d2))*ebz;
  let xx=-i*etb+eps;n0:=n1:=d2:=d3:=0;
    coeff(numer,eps,n);coeff(denom,eps,d);
    res4:=(n1/d2+n0/d2*(dz-d3/d2))/ebz;
out prop;write prop(l,m):=res1+res2+res3+res4;out t;
clear xx
end;
```

The Matrix A

The matrix A as it was calculated in REDUCE2 is given below:

$$A = \begin{bmatrix} 0 & 0 & -\sqrt{-1}\,p & \frac{1}{\beta^2\rho} & 0 & 0 \\ 0 & 0 & -\sqrt{-1}\,q & 0 & \frac{1}{\beta^2\rho} & 0 \\ \frac{\sqrt{-1}\,p(2\beta^2-\alpha^2)}{\alpha^2} & \frac{\sqrt{-1}\,q(2\beta^2-\alpha^2)}{\alpha^2} & 0 & 0 & 0 & \frac{1}{\alpha^2\rho} \\ A_{41} & pq\beta^2\rho(3\alpha^2-4\beta^2)/\alpha^2 & 0 & 0 & 0 & \frac{\sqrt{-1}\,p(2\beta^2-\alpha^2)}{\alpha^2} \\ pq\beta^2\rho(3\alpha^2-4\beta^2)/\alpha^2 & A_{52} & 0 & 0 & 0 & \frac{\sqrt{-1}\,q(2\beta^2-\alpha^2)}{\alpha^2} \\ 0 & 0 & -\omega^2\rho & -\sqrt{-1}\,p & -\sqrt{-1}\,q & 0 \end{bmatrix}$$

where

$$A_{41} = \frac{\rho(4p^2\alpha^2\beta^2 - \omega^2\alpha^2 - 4p^2\beta^4 + q^2\alpha^2\beta^2)}{\alpha^2}$$

$$A_{52} = \frac{\rho(p^2\alpha^2\beta^2 - \omega^2\alpha^2 - 4q^2\beta^4 + 4q^2\alpha^2\beta^2)}{\alpha^2}$$

The Propagator Matrix

Our propagator matrix as derived in REDUCE2 is given below in the form of a FORTRAN subroutine.

```
      SUBROUTINE PROPG2(ALP,BET,RHO,Z,Z0,P,Q,0,PROP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMPLEX*16 I,PROP(6,6),ETA,ETB,EAZ,EBZ,EZ
      REAL*8 K,ALP,BET,RHO,Z,Z0,P,Q,0
      I=(0.0,1.0)
      K=DSQRT(P**2+Q**2)
      DZ=Z-Z0
      ETA = 0**2/ALP**2-K**2
      ETA = CDSQRT(ETA)
      ETB = 0**2/BET**2-K**2
      ETB = CDSQRT(ETB)
      EAZ=CDEXP(I*DZ*ETA)
      EBZ=CDEXP(I*DZ*ETB)
      EZ=EAZ*EBZ
C
      PROP(1,1)=(2*K**4*BET**4*EAZ*EZ-2*K**4*BET**4*EAZ-2*K**4*BET
     !**4*EBZ*EZ+2*K**4*BET**4*EBZ-2*K**2*0**2*BET**2*EAZ*EZ+3*K**
     !2*0**2*BET**2*EAZ+3*K**2*0**2*BET**2*EBZ*EZ-2*K**2*0**2*BET*
     !*2*EBZ-2*K**2*Q**2*BET**4*EAZ*EZ+2*K**2*Q**2*BET**4*EAZ+2*K*
     !*2*Q**2*BET**4*EBZ*EZ-2*K**2*Q**2*BET**4*EBZ-0**4*EAZ-0**4*E
     !BZ*EZ+2*0**2*Q**2*BET**2*EAZ*EZ-2*0**2*Q**2*BET**2*EAZ-2*0**
     !2*Q**2*BET**2*EBZ*EZ+2*0**2*Q**2*BET**2*EBZ)/(2*0**2*EZ*(K**
     !2*BET**2-0**2))
      PROP(1,2)=(P*Q*BET**2*(K**2*BET**2*EAZ*EZ-K**2*BET**2*EAZ-K*
     !*2*BET**2*EBZ*EZ+K**2*BET**2*EBZ-0**2*EAZ*EZ+0**2*EAZ+0**2*E
     !BZ*EZ-0**2*EBZ))/(0**2*EZ*(K**2*BET**2-0**2))
      PROP(1,3)=(P*(-2*K**4*ALP**4*BET**4*EAZ*EZ+2*K**4*ALP**4*BET
     !**4*EBZ+4*K**4*ALP**2*BET**6*EAZ*EZ-4*K**4*ALP**2*BET**6*EBZ
     !-2*K**4*BET**8*EAZ*EZ+2*K**4*BET**8*EBZ+3*K**2*0**2*ALP**4*B
     !ET**2*EAZ*EZ-3*K**2*0**2*ALP**4*BET**2*EBZ-6*K**2*0**2*ALP**
     !2*BET**4*EAZ*EZ-6*K**2*0**2*ALP**2*BET**4*EBZ+3*K**2*0**2*BE
     !T**6*EAZ*EZ-3*K**2*0**2*BET**6*EBZ+2*K**2*ALP**4*ETA*ETB*BET
     !**4*EAZ-2*K**2*ALP**4*ETA*ETB*BET**4*EBZ*EZ-4*K**2*ALP**2*ET
     !A*ETB*BET**6*EAZ+4*K**2*ALP**2*ETA*ETB*BET**6*EBZ*EZ+2*K**2*
     !ETA*ETB*BET**8*EAZ-2*K**2*ETA*ETB*BET**8*EBZ*EZ-0**4*ALP**4*
     !EAZ*EZ+0**4*ALP**4*EBZ+2*0**4*ALP**2*BET**2*EAZ*EZ-2*0**4*AL
     !P**2*BET**2*EBZ-0**4*BET**4*EAZ*EZ+0**4*BET**4*EBZ-2*0**2*AL
     !P**4*ETA*ETB*BET**2*EAZ+2*0**2*ALP**4*ETA*ETB*BET**2*EBZ*EZ+
     !4*0**2*ALP**2*ETA*ETB*BET**4*EAZ-4*0**2*ALP**2*ETA*ETB*BET**
     !4*EBZ*EZ-2*0**2*ETA*ETB*BET**6*EAZ+2*0**2*ETA*ETB*BET**6*EBZ
     !*EZ))/(2*0**2*ETA*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BET**
     !4+K**2*BET**6-0**2*ALP**4+2*0**2*ALP**2*BET**2-0**2*BET**4))
      PROP(1,4)=(K**4*ALP**4*BET**2*EAZ*EZ-K**4*ALP**4*BET**2*EBZ-
     !2*K**4*ALP**2*BET**4*EAZ*EZ+2*K**4*ALP**2*BET**4*EBZ+K**4*BE
     !T**6*EAZ*EZ-K**4*BET**6*EBZ-K**2*0**2*ALP**4*EAZ*EZ+K**2*0**
     !2*ALP**4*EBZ+2*K**2*0**2*ALP**2*BET**2*EAZ*EZ-2*K**2*0**2*AL
```

```
!P**2*BET**2*EBZ-K**2*O**2*BET**4*EAZ*EZ+K**2*O**2*BET**4*EBZ
!-K**2*Q**2*ALP**4*BET**2*EAZ*EZ+K**2*Q**2*ALP**4*BET**2*EBZ+
!2*K**2*Q**2*ALP**2*BET**4*EAZ*EZ-2*K**2*Q**2*ALP**2*BET**4*E
!BZ-K**2*Q**2*BET**6*EAZ*EZ+K**2*Q**2*BET**6*EBZ-K**2*ALP**4*
!ETA*ETB*BET**2*EAZ+K**2*ALP**4*ETA*ETB*BET**2*EBZ*EZ+2*K**2*
!ALP**2*ETA*ETB*BET**4*EAZ-2*K**2*ALP**2*ETA*ETB*BET**4*EBZ*E
!Z-K**2*ETA*ETB*BET**6*EAZ+K**2*ETA*ETB*BET**6*EBZ*EZ+O**2*Q*
!*2*ALP**4*EAZ*EZ-O**2*Q**2*ALP**4*EBZ-2*O**2*Q**2*ALP**2*BET
!**2*EAZ*EZ+2*O**2*Q**2*ALP**2*BET**2*EBZ+O**2*Q**2*BET**4*EA
!Z*EZ-O**2*Q**2*BET**4*EBZ+O**2*ALP**4*ETA*ETB*EAZ-O**2*ALP**
!4*ETA*ETB*EBZ*EZ-2*O**2*ALP**2*ETA*ETB*BET**2*EAZ+2*O**2*ALP
!**2*ETA*ETB*BET**2*EBZ*EZ+O**2*ETA*ETB*BET**4*EAZ-O**2*ETA*E
!TB*BET**4*EBZ*EZ+Q**2*ALP**4*ETA*ETB*BET**2*EAZ-Q**2*ALP**4*
!ETA*ETB*BET**2*EBZ*EZ-2*Q**2*ALP**2*ETA*ETB*BET**4*EAZ+2*Q**
!2*ALP**2*ETA*ETB*BET**4*EBZ*EZ+Q**2*ETA*ETB*BET**6*EAZ-Q**2*
!ETA*ETB*BET**6*EBZ*EZ)
 PROP(1,4)=PROP(1,4)/(2*I*O**2*ETA*RHO*EZ*(K**2*ALP**4*BET
!**2-2*K**2*ALP**2*BET**4+K**2*BET**6-O**2*ALP**4+2*O**2*ALP*
!*2*BET**2-O**2*BET**4))
 PROP(1,5)=(P*Q*(K**2*ALP**4*BET**2*EAZ*EZ-K**2*ALP**4*BET**2
!*EBZ-2*K**2*ALP**2*BET**4*EAZ*EZ+2*K**2*ALP**2*BET**4*EBZ+K*
!*2*BET**6*EAZ*EZ-K**2*BET**6*EBZ-O**2*ALP**4*EAZ*EZ+O**2*ALP
!**4*EBZ+2*O**2*ALP**2*BET**2*EAZ*EZ-2*O**2*ALP**2*BET**2*EBZ
!-O**2*BET**4*EAZ*EZ+O**2*BET**4*EBZ-ALP**4*ETA*ETB*BET**2*EA
!Z+ALP**4*ETA*ETB*BET**2*EBZ*EZ+2*ALP**2*ETA*ETB*BET**4*EAZ-2
!*ALP**2*ETA*ETB*BET**4*EBZ*EZ-ETA*ETB*BET**6*EAZ+ETA*ETB*BET
!**6*EBZ*EZ))/(2*I*O**2*ETA*RHO*EZ*(K**2*ALP**4*BET**2-2*K**2
!*ALP**2*BET**4+K**2*BET**6-O**2*ALP**4+2*O**2*ALP**2*BET**2-
!O**2*BET**4))
 PROP(1,6)=(P*(K**2*ALP**4*BET**2*EAZ*EZ-K**2*ALP**4*BET**2*E
!AZ-K**2*ALP**4*BET**2*EBZ*EZ+K**2*ALP**4*BET**2*EBZ-2*K**2*A
!LP**2*BET**4*EAZ*EZ+2*K**2*ALP**2*BET**4*EAZ+2*K**2*ALP**2*B
!ET**4*EBZ*EZ-2*K**2*ALP**2*BET**4*EBZ+K**2*BET**6*EAZ*EZ-K**
!2*BET**6*EAZ-K**2*BET**6*EBZ*EZ+K**2*BET**6*EBZ-O**2*ALP**4*
!EAZ*EZ+O**2*ALP**4*EAZ+O**2*ALP**4*EBZ*EZ-O**2*ALP**4*EBZ+2*
!O**2*ALP**2*BET**2*EAZ*EZ-2*O**2*ALP**2*BET**2*EAZ-2*O**2*AL
!P**2*BET**2*EBZ*EZ+2*O**2*ALP**2*BET**2*EBZ-O**2*BET**4*EAZ*
!EZ+O**2*BET**4*EAZ+O**2*BET**4*EBZ*EZ-O**2*BET**4*EBZ))/(2*I
!*O**2*RHO*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BET**4+K**2*B
!ET**6-O**2*ALP**4+2*O**2*ALP**2*BET**2-O**2*BET**4))
 PROP(2,1)=(P*Q*BET**2*(K**2*BET**2*EAZ*EZ-K**2*BET**2*EAZ-K*
!*2*BET**2*EBZ*EZ+K**2*BET**2*EBZ-O**2*EAZ*EZ+O**2*EAZ+O**2*E
!BZ*EZ-O**2*EBZ))/(O**2*EZ*(K**2*BET**2-O**2))
 PROP(2,2)=(K**2*O**2*BET**2*EAZ+K**2*O**2*BET**2*EBZ*EZ+2*K*
!*2*Q**2*BET**4*EAZ*EZ-2*K**2*Q**2*BET**4*EAZ-2*K**2*Q**2*BET
!**4*EBZ*EZ+2*K**2*Q**2*BET**4*EBZ-O**4*EAZ-O**4*EBZ*EZ-2*O**
!2*Q**2*BET**2*EAZ*EZ+2*O**2*Q**2*BET**2*EAZ+2*O**2*Q**2*BET*
!*2*EBZ*EZ-2*O**2*Q**2*BET**2*EBZ)/(2*O**2*EZ*(K**2*BET**2-O*
!*2))
 PROP(2,3)=(Q*(-2*K**4*ALP**4*BET**4*EAZ*EZ+2*K**4*ALP**4*BET
!**4*EBZ+4*K**4*ALP**2*BET**6*EAZ*EZ-4*K**4*ALP**2*BET**6*EBZ
!-2*K**4*BET**8*EAZ*EZ+2*K**4*BET**8*EBZ+3*K**2*O**2*ALP**4*B
!ET**2*EAZ*EZ-3*K**2*O**2*ALP**4*BET**2*EBZ-6*K**2*O**2*ALP**
```

```
!2*BET**4*EAZ*EZ+6*K**2*O**2*ALP**2*BET**4*EBZ+3*K**2*O**2*BE
!T**6*EAZ*EZ-3*K**2*O**2*BET**6*EBZ+2*K**2*ALP**4*ETA*ETB*BET
!**4*EAZ-2*K**2*ALP**4*ETA*ETB*BET**4*EBZ*EZ-4*K**2*ALP**2*ET
!A*ETB*BET**6*EAZ+4*K**2*ALP**2*ETA*ETB*BET**6*EBZ*EZ+2*K**2*
!ETA*ETB*BET**8*EAZ-2*K**2*ETA*ETB*BET**8*EBZ*EZ-O**4*ALP**4*
!EAZ*EZ+O**4*ALP**4*EBZ+2*O**4*ALP**2*BET**2*EAZ*EZ-2*O**4*AL
!P**2*BET**2*EBZ-O**4*BET**4*EAZ*EZ+O**4*BET**4*EBZ-2*O**2*AL
!P**4*ETA*ETB*BET**2*EAZ+2*O**2*ALP**4*ETA*ETB*BET**2*EBZ*EZ+
!4*O**2*ALP**2*ETA*ETB*BET**4*EAZ-4*O**2*ALP**2*ETA*ETB*BET**
!4*EBZ*EZ-2*O**2*ETA*ETB*BET**6*EAZ+2*O**2*ETA*ETB*BET**6*EBZ
!*EZ))/(2*O**2*ETA*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BET**
!4+K**2*BET**6-O**2*ALP**4+2*O**2*ALP**2*BET**2-O**2*BET**4))
 PROP(2,4)=(P*Q*(K**2*ALP**4*BET**2*EAZ*EZ-K**2*ALP**4*BET**2
!*EBZ-2*K**2*ALP**2*BET**4*EAZ*EZ+2*K**2*ALP**2*BET**4*EBZ+K*
!*2*BET**6*EAZ*EZ-K**2*BET**6*EBZ-O**2*ALP**4*EAZ*EZ+O**2*ALP
!**4*EBZ+2*O**2*ALP**2*BET**2*EAZ*EZ-2*O**2*ALP**2*BET**2*EBZ
!-O**2*BET**4*EAZ*EZ+O**2*BET**4*EBZ-ALP**4*ETA*ETB*BET**2*EA
!Z+ALP**4*ETA*ETB*BET**2*EBZ*EZ+2*ALP**2*ETA*ETB*BET**4*EAZ-2
!*ALP**2*ETA*ETB*BET**4*EBZ*EZ-ETA*ETB*BET**6*EAZ+ETA*ETB*BET
!**6*EBZ*EZ))/(2*I*O**2*ETA*RHO*EZ*(K**2*ALP**4*BET**2-2*K**2
!*ALP**2*BET**4+K**2*BET**6-O**2*ALP**4+2*O**2*ALP**2*BET**2-
!O**2*BET**4))
 PROP(2,5)=(K**2*Q**2*ALP**4*BET**2*EAZ*EZ-K**2*Q**2*ALP**4*B
!ET**2*EBZ-2*K**2*Q**2*ALP**2*BET**4*EAZ*EZ+2*K**2*Q**2*ALP**
!2*BET**4*EBZ+K**2*Q**2*BET**6*EAZ*EZ-K**2*Q**2*BET**6*EBZ-O*
!*2*Q**2*ALP**4*EAZ*EZ+O**2*Q**2*ALP**4*EBZ+2*O**2*Q**2*ALP**
!2*BET**2*EAZ*EZ-2*O**2*Q**2*ALP**2*BET**2*EBZ-O**2*Q**2*BET*
!*4*EAZ*EZ+O**2*Q**2*BET**4*EBZ+O**2*ALP**4*ETA*ETB*EAZ-O**2*
!ALP**4*ETA*ETB*EBZ*EZ-2*O**2*ALP**2*ETA*ETB*BET**2*EAZ+2*O**
!2*ALP**2*ETA*ETB*BET**2*EBZ*EZ+O**2*ETA*ETB*BET**4*EAZ-O**2*
!ETA*ETB*BET**4*EBZ*EZ-Q**2*ALP**4*ETA*ETB*BET**2*EAZ+Q**2*AL
!P**4*ETA*ETB*BET**2*EBZ*EZ+2*Q**2*ALP**2*ETA*ETB*BET**4*EAZ-
!2*Q**2*ALP**2*ETA*ETB*BET**4*EBZ*EZ-Q**2*ETA*ETB*BET**6*EAZ+
!Q**2*ETA*ETB*BET**6*EBZ*EZ)/(2*I*O**2*ETA*RHO*EZ*(K**2*ALP**
!4*BET**2-2*K**2*ALP**2*BET**4+K**2*BET**6-O**2*ALP**4+2*O**2
!*ALP**2*BET**2-O**2*BET**4))
 PROP(2,6)=(Q*(K**2*ALP**4*BET**2*EAZ*EZ-K**2*ALP**4*BET**2*E
!AZ-K**2*ALP**4*BET**2*EBZ*EZ+K**2*ALP**4*BET**2*EBZ-2*K**2*A
!LP**2*BET**4*EAZ*EZ+2*K**2*ALP**2*BET**4*EAZ+2*K**2*ALP**2*B
!ET**4*EBZ*EZ-2*K**2*ALP**2*BET**4*EBZ+K**2*BET**6*EAZ*EZ-K**
!2*BET**6*EAZ-K**2*BET**6*EBZ*EZ+K**2*BET**6*EBZ-O**2*ALP**4*
!EAZ*EZ+O**2*ALP**4*EAZ+O**2*ALP**4*EBZ*EZ-O**2*ALP**4*EBZ+2*
!O**2*ALP**2*BET**2*EAZ*EZ-2*O**2*ALP**2*BET**2*EAZ-2*O**2*AL
!P**2*BET**2*EBZ*EZ+2*O**2*ALP**2*BET**2*EBZ-O**2*BET**4*EAZ*
!EZ+O**2*BET**4*EAZ+O**2*BET**4*EBZ*EZ-O**2*BET**4*EBZ))/(2*I
!*O**2*RHO*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BET**4+K**2*B
!ET**6-O**2*ALP**4+2*O**2*ALP**2*BET**2-O**2*BET**4))
 PROP(3,1)=(ALP**6*ETA-2*ALP**4*BET**2*ETA+ALP**2*BET**4*ETA
!)**(-1)*P*BET**2*EAZ**(-1)*(-K**2*O**(-2)*ALP**6*EAZ**2+K**2
!*O**(-2)*ALP**6+2*K**2*O**(-2)*ALP**4*BET**2*EAZ**2-2*K**2*O
!**(-2)*ALP**4*BET**2-K**2*O**(-2)*ALP**2*BET**4*EAZ**2+K**2*
!O**(-2)*ALP**2*BET**4+ALP**4*EAZ**2-ALP**4-2*ALP**2*BET**2*E
!AZ**2+2*ALP**2*BET**2+BET**4*EAZ**2-BET**4)+(2*O**4*ALP**2-2
```

```
!*0**4*BET**2-2*0**2*ALP**2*BET**2*K**2+2*0**2*BET**4*K**2)**
!(-1)*P*BET**2*ETB*EBZ**(-1)*(2*K**2*ALP**2*BET**2*EBZ**2-2*K
!**2*ALP**2*BET**2-2*K**2*BET**4*EBZ**2+2*K**2*BET**4-0**2*AL
!P**2*EBZ**2+0**2*ALP**2+0**2*BET**2*EBZ**2-0**2*BET**2)
 PROP(3,2)=(Q*BET**2*(-2*K**4*ALP**8*BET**2*EAZ*EZ+2*K**4*ALP
!**8*BET**2*EBZ+6*K**4*ALP**6*BET**4*EAZ*EZ-6*K**4*ALP**6*BET
!**4*EBZ-6*K**4*ALP**4*BET**6*EAZ*EZ+6*K**4*ALP**4*BET**6*EBZ
!+2*K**4*ALP**2*BET**8*EAZ*EZ-2*K**4*ALP**2*BET**8*EBZ+2*K**2
!*0**2*ALP**8*EAZ*EZ-2*K**2*0**2*ALP**8*EBZ-4*K**2*0**2*ALP**
!6*BET**2*EAZ*EZ+4*K**2*0**2*ALP**6*BET**2*EBZ+4*K**2*0**2*AL
!P**2*BET**6*EAZ*EZ-4*K**2*0**2*ALP**2*BET**6*EBZ-2*K**2*0**2
!*BET**8*EAZ*EZ+2*K**2*0**2*BET**8*EBZ+2*K**2*ALP**8*ETA*ETB*
!BET**2*EAZ-2*K**2*ALP**8*ETA*ETB*BET**2*EBZ*EZ-6*K**2*ALP**6
!*ETA*ETB*BET**4*EAZ+6*K**2*ALP**6*ETA*ETB*BET**4*EBZ*EZ+6*K*
!*2*ALP**4*ETA*ETB*BET**6*EAZ-6*K**2*ALP**4*ETA*ETB*BET**6*EB
!Z*EZ-2*K**2*ALP**2*ETA*ETB*BET**8*EAZ+2*K**2*ALP**2*ETA*ETB*
!BET**8*EBZ*EZ-2*0**4*ALP**6*EAZ*EZ+2*0**4*ALP**6*EBZ+6*0**4*
!ALP**4*BET**2*EAZ*EZ-6*0**4*ALP**4*BET**2*EBZ-6*0**4*ALP**2*
!BET**4*EAZ*EZ+6*0**4*ALP**2*BET**4*EBZ+2*0**4*BET**6*EAZ*EZ-
!2*0**4*BET**6*EBZ-0**2*ALP**8*ETA*ETB*EAZ+0**2*ALP**8*ETA*ET
!B*EBZ*EZ+3*0**2*ALP**6*ETA*ETB*BET**2*EAZ-3*0**2*ALP**6*ETA*
!ETB*BET**2*EBZ*EZ-3*0**2*ALP**4*ETA*ETB*BET**4*EAZ+3*0**2*AL
!P**4*ETA*ETB*BET**4*EBZ*EZ+0**2*ALP**2*ETA*ETB*BET**6*EAZ-0*
!*2*ALP**2*ETA*ETB*BET**6*EBZ*EZ))
 PROP(3,2)=PROP(3,2)/(2*0**2*ALP**2*ETA*EZ*(K**
!2*ALP**6*BET**2-3*K**2*ALP**4*BET**4+3*K**2*ALP**2*BET**6-K*
!*2*BET**8-0**2*ALP**6+3*0**2*ALP**4*BET**2-3*0**2*ALP**2*BET
!**4+0**2*BET**6))
 PROP(3,3)=(-2*K**4*BET**4*EAZ*EZ+2*K**4*BET**4*EAZ+2*K**4*BE
!T**4*EBZ*EZ-2*K**4*BET**4*EBZ+3*K**2*0**2*BET**2*EAZ*EZ-2*K*
!*2*0**2*BET**2*EAZ-2*K**2*0**2*BET**2*EBZ*EZ+3*K**2*0**2*BET
!**2*EBZ-0**4*EAZ*EZ-0**4*EBZ)/(2*0**2*EZ*(K**2*BET**2-0**2))
 PROP(3,4)=(P*(K**2*ALP**4*BET**2*EAZ*EZ-K**2*ALP**4*BET**2*E
!AZ-K**2*ALP**4*BET**2*EBZ*EZ+K**2*ALP**4*BET**2*EBZ-2*K**2*A
!LP**2*BET**4*EAZ*EZ+2*K**2*ALP**2*BET**4*EAZ+2*K**2*ALP**2*B
!ET**4*EBZ*EZ-2*K**2*ALP**2*BET**4*EBZ+K**2*BET**6*EAZ*EZ-K**
!2*BET**6*EAZ-K**2*BET**6*EBZ*EZ+K**2*BET**6*EBZ-0**2*ALP**4*
!EAZ*EZ+0**2*ALP**4*EAZ+0**2*ALP**4*EBZ*EZ-0**2*ALP**4*EBZ+2*
!0**2*ALP**2*BET**2*EAZ*EZ-2*0**2*ALP**2*BET**2*EAZ-2*0**2*AL
!P**2*BET**2*EBZ*EZ+2*0**2*ALP**2*BET**2*EBZ-0**2*BET**4*EAZ*
!EZ+0**2*BET**4*EAZ+0**2*BET**4*EBZ*EZ-0**2*BET**4*EBZ))/(2*I
!*0**2*RH0*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BET**4+K**2*B
!ET**6-0**2*ALP**4+2*0**2*ALP**2*BET**2-0**2*BET**4))
 PROP(3,5)=(Q*(K**2*ALP**4*BET**2*EAZ*EZ-K**2*ALP**4*BET**2*E
!AZ-K**2*ALP**4*BET**2*EBZ*EZ+K**2*ALP**4*BET**2*EBZ-2*K**2*A
!LP**2*BET**4*EAZ*EZ+2*K**2*ALP**2*BET**4*EAZ+2*K**2*ALP**2*B
!ET**4*EBZ*EZ-2*K**2*ALP**2*BET**4*EBZ+K**2*BET**6*EAZ*EZ-K**
!2*BET**6*EAZ-K**2*BET**6*EBZ*EZ+K**2*BET**6*EBZ-0**2*ALP**4*
!EAZ*EZ+0**2*ALP**4*EAZ+0**2*ALP**4*EBZ*EZ-0**2*ALP**4*EBZ+2*
!0**2*ALP**2*BET**2*EAZ*EZ-2*0**2*ALP**2*BET**2*EAZ-2*0**2*AL
!P**2*BET**2*EBZ*EZ+2*0**2*ALP**2*BET**2*EBZ-0**2*BET**4*EAZ*
!EZ+0**2*BET**4*EAZ+0**2*BET**4*EBZ*EZ-0**2*BET**4*EBZ))/(2*I
!*0**2*RH0*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BET**4+K**2*B
!ET**6-0**2*ALP**4+2*0**2*ALP**2*BET**2-0**2*BET**4))
```

```
 PROP(3,6)=(-K**4*ALP**2*BET**2*EAZ*EZ+K**4*ALP**2*BET**2*E
!BZ+K**2*O**2*ALP**2*EAZ*EZ-K**2*O**2*ALP**2*EBZ+K**2*O**2*
!BET**2*EAZ*EZ-K**2*O**2*BET**2*EBZ+K**2*ALP**2*ETA*ETB*BET
!**2*EAZ-K**2*ALP**2*ETA*ETB*BET**2*EBZ*EZ-O**4*EAZ*EZ+O**4
!*EBZ)/(2*I*O**2*ALP**2*ETA*RHO*EZ*(K**2*BET**2-O**2))
 PROP(4,1)=(2*RHO*BET**4*(K**4*ALP**6*EAZ**2-K**4*ALP**6-2*K*
!*4*ALP**4*BET**2*EAZ**2+2*K**4*ALP**4*BET**2+K**4*ALP**2*BET
!**4*EAZ**2-K**4*ALP**2*BET**4-K**2*O**2*ALP**4*EAZ**2+K**2*O
!**2*ALP**4+2*K**2*O**2*ALP**2*BET**2*EAZ**2-2*K**2*O**2*ALP*
!*2*BET**2-K**2*O**2*BET**4*EAZ**2+K**2*O**2*BET**4-K**2*Q**2
!*ALP**6*EAZ**2+K**2*Q**2*ALP**6+2*K**2*Q**2*ALP**4*BET**2*EA
!Z**2-2*K**2*Q**2*ALP**4*BET**2-K**2*Q**2*ALP**2*BET**4*EAZ**
!2+K**2*Q**2*ALP**2*BET**4+O**2*Q**2*ALP**4*EAZ**2-O**2*Q**2*
!ALP**4-2*O**2*Q**2*ALP**2*BET**2*EAZ**2+2*O**2*Q**2*ALP**2*B
!ET**2+O**2*Q**2*BET**4*EAZ**2-O**2*Q**2*BET**4))/(I*O**2*ALP
!**2*EAZ*ETA*(ALP**4-2*ALP**2*BET**2+BET**4))+(I*RHO*BET**2*E
!TB*(-4*K**4*ALP**2*BET**4*EBZ**2+4*K**4*ALP**2*BET**4+4*K**4
!*BET**6*EBZ**2-4*K**4*BET**6+4*K**2*O**2*ALP**2*BET**2*EBZ**
!2-4*K**2*O**2*ALP**2*BET**2-4*K**2*O**2*BET**4*EBZ**2+4*K**2
!*O**2*BET**4+4*K**2*Q**2*ALP**2*BET**4*EBZ**2-4*K**2*Q**2*AL
!P**2*BET**4-4*K**2*Q**2*BET**6*EBZ**2+4*K**2*Q**2*BET**6-O**
!4*ALP**2*EBZ**2+O**4*ALP**2+O**4*BET**2*EBZ**2-O**4*BET**2-3
!*O**2*Q**2*ALP**2*BET**2*EBZ**2+3*O**2*Q**2*ALP**2*BET**2+3*
!O**2*Q**2*BET**4*EBZ**2-3*O**2*Q**2*BET**4))/(2*O**2*EBZ*(K*
!*2*ALP**2*BET**2-K**2*BET**4-O**2*ALP**2+O**2*BET**2))
 PROP(4,2)=(2*P*Q*RHO*BET**4*(K**2*EAZ**2*ALP**6-2*K**2*EAZ**
!2*ALP**4*BET**2+K**2*EAZ**2*ALP**2*BET**4-K**2*ALP**6+2*K**2
!*ALP**4*BET**2-K**2*ALP**2*BET**4-O**2*EAZ**2*ALP**4+2*O**2*
!EAZ**2*ALP**2*BET**2-O**2*EAZ**2*BET**4+O**2*ALP**4-2*O**2*A
!LP**2*BET**2+O**2*BET**4))/(I*O**2*EAZ*ALP**2*ETA*(ALP**4-2*
!ALP**2*BET**2+BET**4))+(I*P*Q*RHO*ETB*BET**4*(4*K**2*EBZ**2*
!ALP**2*BET**2-4*K**2*EBZ**2*BET**4-4*K**2*ALP**2*BET**2+4*K*
!*2*BET**4-3*O**2*EBZ**2*ALP**2+3*O**2*EBZ**2*BET**2+3*O**2*A
!LP**2-3*O**2*BET**2))/(2*O**2*EBZ*(-K**2*ALP**2*BET**2+K**2*
!BET**4+O**2*ALP**2-O**2*BET**2))
 PROP(4,3)=(P*BET**2*RHO*(2*K**4*ALP**4*BET**4*EAZ*EZ-2*K**
!4*ALP**4*BET**4*EAZ-2*K**4*ALP**4*BET**4*EBZ*EZ+2*K**4*ALP
!**4*BET**4*EBZ-4*K**4*ALP**2*BET**6*EAZ*EZ+4*K**4*ALP**2*B
!ET**6*EAZ+4*K**4*ALP**2*BET**6*EBZ*EZ-4*K**4*ALP**2*BET**6
!*EBZ+2*K**4*BET**8*EAZ*EZ-2*K**4*BET**8*EAZ-2*K**4*BET**8*
!EBZ*EZ+2*K**4*BET**8*EBZ-3*K**2*O**2*ALP**4*BET**2*EAZ*EZ+3*
!K**2*O**2*ALP**4*BET**2*EAZ+3*K**2*O**2*ALP**4*BET**2*EBZ*
!EZ-3*K**2*O**2*ALP**4*BET**2*EBZ+6*K**2*O**2*ALP**2*BET**4
!*EAZ*EZ-6*K**2*O**2*ALP**2*BET**4*EAZ-6*K**2*O**2*ALP**2*B
!ET**4*EBZ*EZ+6*K**2*O**2*ALP**2*BET**4*EBZ-3*K**2*O**2*BET
!**6*EAZ*EZ+3*K**2*O**2*BET**6*EAZ+3*K**2*O**2*BET**6*EBZ*E
!Z-3*K**2*O**2*BET**6*EBZ+O**4*ALP**4*EAZ*EZ-O**4*ALP**4*EAZ-
!O**4*ALP**4*EBZ*EZ+O**4*ALP**4*EBZ-2*O**4*ALP**2*BET**2*EA
!Z*EZ+2*O**4*ALP**2*BET**2*EAZ+2*O**4*ALP**2*BET**2*EBZ*EZ-
!2*O**4*ALP**2*BET**2*EBZ+O**4*BET**4*EAZ*EZ-O**4*BET**4*EA
!Z-O**4*BET**4*EBZ*EZ+O**4*BET**4*EBZ))/(I*O**2*EZ*(K**2*AL
!P**4*BET**2-2*K**2*ALP**2*BET**4+K**2*BET**6-O**2*ALP**4+2
!*O**2*ALP**2*BET**2-O**2*BET**4))
```

```
 PROP(4,4)=(2*K**4*BET**4*EAZ*EZ-2*K**4*BET**4*EAZ-2*K**4*B
!ET**4*EBZ*EZ+2*K**4*BET**4*EBZ-2*K**2*O**2*BET**2*EAZ*EZ+3
!*K**2*O**2*BET**2*EAZ+3*K**2*O**2*BET**2*EBZ*EZ-2*K**2*O**
!2*BET**2*EBZ-2*K**2*Q**2*BET**4*EAZ*EZ+2*K**2*Q**2*BET**4*
!EAZ+2*K**2*Q**2*BET**4*EBZ*EZ-2*K**2*Q**2*BET**4*EBZ-O**4*
!EAZ-O**4*EBZ*EZ+2*O**2*Q**2*BET**2*EAZ*EZ-2*O**2*Q**2*BET**2
!*EAZ-2*O**2*Q**2*BET**2*EBZ*EZ+2*O**2*Q**2*BET**2*EBZ)/(2*
!O**2*EZ*(K**2*BET**2-O**2))
 PROP(4,5)=(P*Q*BET**2*(K**2*BET**2*EAZ*EZ-K**2*BET**2*EAZ-
!K**2*BET**2*EBZ*EZ+K**2*BET**2*EBZ-O**2*EAZ*EZ+O**2*EAZ+O*
!*2*EBZ*EZ-O**2*EBZ))/(O**2*EZ*(K**2*BET**2-O**2))
 PROP(4,6)=(P*BET**2*(-K**2*EAZ**2*ALP**6+2*K**2*EAZ**2*ALP**
!4*BET**2-K**2*EAZ**2*ALP**2*BET**4+K**2*ALP**6-2*K**2*ALP**4
!*BET**2+K**2*ALP**2*BET**4+O**2*EAZ**2*ALP**4-2*O**2*EAZ**2*
!ALP**2*BET**2+O**2*EAZ**2*BET**4-O**2*ALP**4+2*O**2*ALP**2*B
!ET**2-O**2*BET**4))/(O**2*ETA*EAZ*ALP**2*(ALP**4-2*ALP**2*BE
!T**2+BET**4))+(P*BET**2*ETB*(2*K**2*EBZ**2*ALP**2*BET**2-2*K
!**2*EBZ**2*BET**4-2*K**2*ALP**2*BET**2+2*K**2*BET**4-O**2*EB
!Z**2*ALP**2+O**2*EBZ**2*BET**2+O**2*ALP**2-O**2*BET**2))/(2*
!O**2*EBZ*(-K**2*ALP**2*BET**2+K**2*BET**4+O**2*ALP**2-O**2*B
!ET**2))
 PROP(5,1)=(2*P*Q*RHO*BET**4*(K**2*EAZ**2*ALP**6-2*K**2*EAZ**
!2*ALP**4*BET**2+K**2*EAZ**2*ALP**2*BET**4-K**2*ALP**6+2*K**2
!*ALP**4*BET**2-K**2*ALP**2*BET**4-O**2*EAZ**2*ALP**4+2*O**2*
!EAZ**2*ALP**2*BET**2-O**2*EAZ**2*BET**4+O**2*ALP**4-2*O**2*A
!LP**2*BET**2+O**2*BET**4))/(I*O**2*EAZ*ALP**2*ETA*(ALP**4-2*
!ALP**2*BET**2+BET**4))+(I*P*Q*RHO*ETB*BET**4*(4*K**2*EBZ**2*
!ALP**2*BET**2-4*K**2*EBZ**2*BET**4-4*K**2*ALP**2*BET**2+4*K*
!*2*BET**4-3*O**2*EBZ**2*ALP**2+3*O**2*EBZ**2*BET**2+3*O**2*A
!LP**2-3*O**2*BET**2))/(2*O**2*EBZ*(-K**2*ALP**2*BET**2+K**2*
!BET**4+O**2*ALP**2-O**2*BET**2))
 PROP(5,2)=(2*Q**2*RHO*BET**4*(K**2*ALP**6*EAZ**2-K**2*ALP**6
!-2*K**2*ALP**4*BET**2*EAZ**2+2*K**2*ALP**4*BET**2+K**2*ALP**
!2*BET**4*EAZ**2-K**2*ALP**2*BET**4-O**2*ALP**4*EAZ**2+O**2*A
!LP**4+2*O**2*ALP**2*BET**2*EAZ**2-2*O**2*ALP**2*BET**2-O**2*
!BET**4*EAZ**2+O**2*BET**4))/(I*O**2*ALP**2*EAZ*ETA*(ALP**4-2
!*ALP**2*BET**2+BET**4))+(I*RHO*BET**2*ETB*(-K**2*O**2*ALP**2
!*BET**2*EBZ**2+K**2*O**2*ALP**2*BET**2+K**2*O**2*BET**4*EBZ*
!*2-K**2*O**2*BET**4+4*K**2*Q**2*ALP**2*BET**4*EBZ**2-4*K**2*
!Q**2*ALP**2*BET**4-4*K**2*Q**2*BET**6*EBZ**2+4*K**2*Q**2*BET
!**6+O**4*ALP**2*EBZ**2-O**4*ALP**2-O**4*BET**2*EBZ**2+O**4*B
!ET**2-3*O**2*Q**2*ALP**2*BET**2*EBZ**2+3*O**2*Q**2*ALP**2*BE
!T**2+3*O**2*Q**2*BET**4*EBZ**2-3*O**2*Q**2*BET**4))/(2*O**2*
!EBZ*(-K**2*ALP**2*BET**2+K**2*BET**4+O**2*ALP**2-O**2*BET**2
!))
 PROP(5,3)=(Q*BET**2*RHO*(2*K**4*ALP**4*BET**4*EAZ*EZ-2*K**
!4*ALP**4*BET**4*EAZ-2*K**4*ALP**4*BET**4*EBZ*EZ+2*K**4*ALP
!**4*BET**4*EBZ-4*K**4*ALP**2*BET**6*EAZ*EZ+4*K**4*ALP**2*B
!ET**6*EAZ+4*K**4*ALP**2*BET**6*EBZ*EZ-4*K**4*ALP**2*BET**6
!*EBZ+2*K**4*BET**8*EAZ*EZ-2*K**4*BET**8*EAZ-2*K**4*BET**8*
!EBZ*EZ+2*K**4*BET**8*EBZ-3*K**2*O**2*ALP**4*BET**2*EAZ*EZ+3*
!K**2*O**2*ALP**4*BET**2*EAZ+3*K**2*O**2*ALP**4*BET**2*EBZ*
!EZ-3*K**2*O**2*ALP**4*BET**2*EBZ+6*K**2*O**2*ALP**2*BET**4
```

```
!*EAZ*EZ-6*K**2*O**2*ALP**2*BET**4*EAZ-6*K**2*O**2*ALP**2*B
!ET**4*EBZ*EZ+6*K**2*O**2*ALP**2*BET**4*EBZ-3*K**2*O**2*BET
!**6*EAZ*EZ+3*K**2*O**2*BET**6*EAZ+3*K**2*O**2*BET**6*EBZ*E
!Z-3*K**2*O**2*BET**6*EBZ+O**4*ALP**4*EAZ*EZ-O**4*ALP**4*EAZ-
!O**4*ALP**4*EBZ*EZ+O**4*ALP**4*EBZ-2*O**4*ALP**2*BET**2*EA
!Z*EZ+2*O**4*ALP**2*BET**2*EAZ+2*O**4*ALP**2*BET**2*EBZ*EZ-
!2*O**4*ALP**2*BET**2*EBZ+O**4*BET**4*EAZ*EZ-O**4*BET**4*EA
!Z-O**4*BET**4*EBZ*EZ+O**4*BET**4*EBZ))/(1*O**2*EZ*(K**2*AL
!P**4*BET**2-2*K**2*ALP**2*BET**4+K**2*BET**6-O**2*ALP**4+2
!*O**2*ALP**2*BET**2-O**2*BET**4))
 PROP(5,4)=(P*Q*BET**2*(K**2*BET**2*EAZ*EZ-K**2*BET**2*EAZ-
!K**2*BET**2*EBZ*EZ+K**2*BET**2*EBZ-O**2*EAZ*EZ+O**2*EAZ+O*
!*2*EBZ*EZ-O**2*EBZ))/(O**2*EZ*(K**2*BET**2-O**2))
 PROP(5,5)=(K**2*O**2*BET**2*EAZ+K**2*O**2*BET**2*EBZ*EZ+2*K*
!*2*Q**2*BET**4*EAZ*EZ-2*K**2*Q**2*BET**4*EAZ-2*K**2*Q**2*BET
!**4*EBZ*EZ+2*K**2*Q**2*BET**4*EBZ-O**4*EAZ-O**4*EBZ*EZ-2*O**
!2*Q**2*BET**2*EAZ*EZ+2*O**2*Q**2*BET**2*EAZ+2*O**2*Q**2*BET*
!*2*EBZ*EZ-2*O**2*Q**2*BET**2*EBZ)/(2*O**2*EZ*(K**2*BET**2-O*
!*2))
 PROP(5,6)=(Q*BET**2*(-K**2*EAZ**2*ALP**6+2*K**2*EAZ**2*ALP**
!4*BET**2-K**2*EAZ**2*ALP**2*BET**4+K**2*ALP**6-2*K**2*ALP**4
!*BET**2+K**2*ALP**2*BET**4+O**2*EAZ**2*ALP**4-2*O**2*EAZ**2*
!ALP**2*BET**2+O**2*EAZ**2*BET**4-O**2*ALP**4+2*O**2*ALP**2*B
!ET**2-O**2*BET**4))/(O**2*ETA*EAZ*ALP**2*(ALP**4-2*ALP**2*BE
!T**2+BET**4))+(Q*BET**2*ETB*(2*K**2*EBZ**2*ALP**2*BET**2-2*K
!**2*EBZ**2*BET**4-2*K**2*ALP**2*BET**2+2*K**2*BET**4-O**2*EB
!Z**2*ALP**2+O**2*EBZ**2*BET**2+O**2*ALP**2-O**2*BET**2))/(2*
!O**2*EBZ*(-K**2*ALP**2*BET**2+K**2*BET**4+O**2*ALP**2-O**2*B
!ET**2))
 PROP(6,1)=(P*BET**2*RHO*(2*K**4*ALP**4*BET**4*EAZ*EZ-2*K**4*
!ALP**4*BET**4*EAZ-2*K**4*ALP**4*BET**4*EBZ*EZ+2*K**4*ALP**4*
!BET**4*EBZ-4*K**4*ALP**2*BET**6*EAZ*EZ+4*K**4*ALP**2*BET**6*
!EAZ+4*K**4*ALP**2*BET**6*EBZ*EZ-4*K**4*ALP**2*BET**6*EBZ+2*K
!**4*BET**8*EAZ*EZ-2*K**4*BET**8*EAZ-2*K**4*BET**8*EBZ*EZ+2*K
!**4*BET**8*EBZ-3*K**2*O**2*ALP**4*BET**2*EAZ*EZ+3*K**2*O**2*
!ALP**4*BET**2*EAZ+3*K**2*O**2*ALP**4*BET**2*EBZ*EZ-3*K**2*O*
!*2*ALP**4*BET**2*EBZ+6*K**2*O**2*ALP**2*BET**4*EAZ*EZ-6*K**2
!*O**2*ALP**2*BET**4*EAZ-6*K**2*O**2*ALP**2*BET**4*EBZ*EZ+6*K
!**2*O**2*ALP**2*BET**4*EBZ-3*K**2*O**2*BET**6*EAZ*EZ+3*K**2*
!O**2*BET**6*EAZ+3*K**2*O**2*BET**6*EBZ*EZ-3*K**2*O**2*BET**6
!*EBZ+O**4*ALP**4*EAZ*EZ-O**4*ALP**4*EAZ-O**4*ALP**4*EBZ*EZ+O
!**4*ALP**4*EBZ-2*O**4*ALP**2*BET**2*EAZ*EZ+2*O**4*ALP**2*BET
!**2*EAZ+2*O**4*ALP**2*BET**2*EBZ*EZ-2*O**4*ALP**2*BET**2*EBZ
!+O**4*BET**4*EAZ*EZ-O**4*BET**4*EAZ-O**4*BET**4*EBZ*EZ+O**4*
!BET**4*EBZ))/(1*O**2*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BE
!T**4+K**2*BET**6-O**2*ALP**4+2*O**2*ALP**2*BET**2-O**2*BET**
!4))
 PROP(6,2)=(Q*BET**2*RHO*(2*K**4*ALP**4*BET**4*EAZ*EZ-2*K**4*
!ALP**4*BET**4*EAZ-2*K**4*ALP**4*BET**4*EBZ*EZ+2*K**4*ALP**4*
!BET**4*EBZ-4*K**4*ALP**2*BET**6*EAZ*EZ+4*K**4*ALP**2*BET**6*
!EAZ+4*K**4*ALP**2*BET**6*EBZ*EZ-4*K**4*ALP**2*BET**6*EBZ+2*K
!**4*BET**8*EAZ*EZ-2*K**4*BET**8*EAZ-2*K**4*BET**8*EBZ*EZ+2*K
!**4*BET**8*EBZ-3*K**2*O**2*ALP**4*BET**2*EAZ*EZ+3*K**2*O**2*
```

```
!ALP**4*BET**2*EAZ+3*K**2*0**2*ALP**4*BET**2*EBZ*EZ-3*K**2*0*
!*2*ALP**4*BET**2*EBZ+6*K**2*0**2*ALP**2*BET**4*EAZ*EZ-6*K**2
!*0**2*ALP**2*BET**4*EAZ-6*K**2*0**2*ALP**2*BET**4*EBZ*EZ+6*K
!**2*0**2*ALP**2*BET**4*EBZ-3*K**2*0**2*BET**6*EAZ*EZ+3*K**2*
!0**2*BET**6*EAZ+3*K**2*0**2*BET**6*EBZ*EZ-3*K**2*0**2*BET**6
!*EBZ+0**4*ALP**4*EAZ*EZ-0**4*ALP**4*EAZ-0**4*ALP**4*EBZ*EZ+0
!**4*ALP**4*EBZ-2*0**4*ALP**2*BET**2*EAZ*EZ+2*0**4*ALP**2*BET
!**2*EAZ+2*0**4*ALP**2*BET**2*EBZ*EZ-2*0**4*ALP**2*BET**2*EBZ
!+0**4*BET**4*EAZ*EZ-0**4*BET**4*EAZ-0**4*BET**4*EBZ*EZ+0**4*
!BET**4*EBZ))/(1*0**2*EZ*(K**2*ALP**4*BET**2-2*K**2*ALP**2*BE
!T**4+K**2*BET**6-0**2*ALP**4+2*0**2*ALP**2*BET**2-0**2*BET**
!4))
 PROP(6,3)=(RHO*(-4*K**6*ALP**4*BET**6*EBZ+4*K**6*ALP**4*BET*
!*6*EAZ*EZ+8*K**6*ALP**2*BET**8*EBZ-8*K**6*ALP**2*BET**8*EAZ*
!EZ-4*K**6*BET**10*EBZ+4*K**6*BET**10*EAZ*EZ+8*K**4*0**2*ALP*
!*4*BET**4*EBZ-8*K**4*0**2*ALP**4*BET**4*EAZ*EZ-16*K**4*0**2*
!ALP**2*BET**6*EBZ+16*K**4*0**2*ALP**2*BET**6*EAZ*EZ+8*K**4*0
!**2*BET**8*EBZ-8*K**4*0**2*BET**8*EAZ*EZ+4*K**4*ALP**4*BET**
!6*ETB*ETA*EBZ*EZ-4*K**4*ALP**4*BET**6*ETB*ETA*EAZ-8*K**4*ALP
!**2*BET**8*ETB*ETA*EBZ*EZ+8*K**4*ALP**2*BET**8*ETB*ETA*EAZ+4
!*K**4*BET**10*ETB*ETA*EBZ*EZ-4*K**4*BET**10*ETB*ETA*EAZ-5*K*
!*2*0**4*ALP**4*BET**2*EBZ+5*K**2*0**4*ALP**4*BET**2*EAZ*EZ+1
!0*K**2*0**4*ALP**2*BET**4*EBZ-10*K**2*0**4*ALP**2*BET**4*EAZ
!*EZ-5*K**2*0**4*BET**6*EBZ+5*K**2*0**4*BET**6*EAZ*EZ-4*K**2*
!0**2*ALP**4*BET**4*ETB*ETA*EBZ*EZ+4*K**2*0**2*ALP**4*BET**4*
!ETB*ETA*EAZ+8*K**2*0**2*ALP**2*BET**6*ETB*ETA*EBZ*EZ-8*K**2*
!0**2*ALP**2*BET**6*ETB*ETA*EAZ-4*K**2*0**2*BET**8*ETB*ETA*EB
!Z*EZ+4*K**2*0**2*BET**8*ETB*ETA*EAZ+0**6*ALP**4*EBZ-0**6*ALP
!**4*EAZ*EZ-2*0**6*ALP**2*BET**2*EBZ+2*0**6*ALP**2*BET**2*EAZ
!*EZ+0**6*BET**4*EBZ-0**6*BET**4*EAZ*EZ))/(2*1*0**2*ETA*EZ*(-
!K**2*ALP**4*BET**2+2*K**2*ALP**2*BET**4-K**2*BET**6+0**2*ALP
!**4-2*0**2*ALP**2*BET**2+0**2*BET**4))
 PROP(6,4)=(P*(-2*K**4*ALP**4*BET**4*EBZ+2*K**4*ALP**4*BET**4
!*EAZ*EZ+4*K**4*ALP**2*BET**6*EBZ-4*K**4*ALP**2*BET**6*EAZ*EZ
!-2*K**4*BET**8*EBZ+2*K**4*BET**8*EAZ*EZ+3*K**2*0**2*ALP**4*B
!ET**2*EBZ-3*K**2*0**2*ALP**4*BET**2*EAZ*EZ-6*K**2*0**2*ALP**
!2*BET**4*EBZ+6*K**2*0**2*ALP**2*BET**4*EAZ*EZ+3*K**2*0**2*BE
!T**6*EBZ-3*K**2*0**2*BET**6*EAZ*EZ+2*K**2*ALP**4*BET**4*ETB*
!ETA*EBZ*EZ-2*K**2*ALP**4*BET**4*ETB*ETA*EAZ-4*K**2*ALP**2*BE
!T**6*ETB*ETA*EBZ*EZ+4*K**2*ALP**2*BET**6*ETB*ETA*EAZ+2*K**2*
!BET**8*ETB*ETA*EBZ*EZ-2*K**2*BET**8*ETB*ETA*EAZ-0**4*ALP**4*
!EBZ+0**4*ALP**4*EAZ*EZ+2*0**4*ALP**2*BET**2*EBZ-2*0**4*ALP**
!2*BET**2*EAZ*EZ-0**4*BET**4*EBZ+0**4*BET**4*EAZ*EZ-2*0**2*AL
!P**4*BET**2*ETB*ETA*EBZ*EZ+2*0**2*ALP**4*BET**2*ETB*ETA*EAZ+
!4*0**2*ALP**2*BET**4*ETB*ETA*EBZ*EZ-4*0**2*ALP**2*BET**4*ETB
!*ETA*EAZ-2*0**2*BET**6*ETB*ETA*EBZ*EZ+2*0**2*BET**6*ETB*ETA*
!EAZ))/(2*0**2*ETA*EZ*(-K**2*ALP**4*BET**2+2*K**2*ALP**2*BET*
!*4-K**2*BET**6+0**2*ALP**4-2*0**2*ALP**2*BET**2+0**2*BET**4))
 PROP(6,5)=(Q*(-2*K**4*ALP**4*BET**4*EBZ+2*K**4*ALP**4*BET**4
!*EAZ*EZ+4*K**4*ALP**2*BET**6*EBZ-4*K**4*ALP**2*BET**6*EAZ*EZ
!-2*K**4*BET**8*EBZ+2*K**4*BET**8*EAZ*EZ+3*K**2*0**2*ALP**4*B
!ET**2*EBZ-3*K**2*0**2*ALP**4*BET**2*EAZ*EZ-6*K**2*0**2*ALP**
!2*BET**4*EBZ+6*K**2*0**2*ALP**2*BET**4*EAZ*EZ+3*K**2*0**2*BE
```

```
     !T**6*EBZ-3*K**2*O**2*BET**6*EAZ*EZ+2*K**2*ALP**4*BET**4*ETB*
     !ETA*EBZ*EZ-2*K**2*ALP**4*BET**4*ETB*ETA*EAZ-4*K**2*ALP**2*BE
     !T**6*ETB*ETA*EBZ*EZ+4*K**2*ALP**2*BET**6*ETB*ETA*EAZ+2*K**2*
     !BET**8*ETB*ETA*EBZ*EZ-2*K**2*BET**8*ETB*ETA*EAZ-O**4*ALP**4*
     !EBZ+O**4*ALP**4*EAZ*EZ+2*O**4*ALP**2*BET**2*EBZ-2*O**4*ALP**
     !2*BET**2*EAZ*EZ-O**4*BET**4*EBZ+O**4*BET**4*EAZ*EZ-2*O**2*AL
     !P**4*BET**2*ETB*ETA*EBZ*EZ+2*O**2*ALP**4*BET**2*ETB*ETA*EAZ+
     !4*O**2*ALP**2*BET**4*ETB*ETA*EBZ*EZ-4*O**2*ALP**2*BET**4*ETB
     !*ETA*EAZ-2*O**2*BET**6*ETB*ETA*EBZ*EZ+2*O**2*BET**6*ETB*ETA*
     !EAZ))/(2*O**2*ETA*EZ*(-K**2*ALP**4*BET**2+2*K**2*ALP**2*BET*
     !*4-K**2*BET**6+O**2*ALP**4-2*O**2*ALP**2*BET**2+O**2*BET**4))
      PROP(6,6)=(-2*K**4*BET**4*EBZ*EZ+2*K**4*BET**4*EBZ+2*K**4*BE
     !T**4*EAZ*EZ-2*K**4*BET**4*EAZ+2*K**2*O**2*BET**2*EBZ*EZ-3*K*
     !*2*O**2*BET**2*EBZ-3*K**2*O**2*BET**2*EAZ*EZ+2*K**2*O**2*BET
     !**2*EAZ+O**4*EBZ+O**4*EAZ*EZ)/(2*O**2*EZ*(-K**2*BET**2+O**2))
      RETURN
      END
$ENDFILE
```

Appendix 2

Computer Code for Production of Synthetic Seismograms

```
C THIS FORTRAN IV PROGRAM CALCULATES SYNTHETIC SEISMOGRAMS FOR A STACK
C OF HOMOGENEOUS LAYERS USING THE PROPAGATOR MATRIX METHOD. THERE IS AN
C INTERNAL POINT MICROSEISMIC SOURCE. MAXIMUM OF FIVE LAYERS
C
C INPUT ON UNIT 5: N:       THE NUMBER OF LAYERS INCLUDING THE HALF-SPACE
C                           (THE HALF-SPACE IS LAYER 1)
C                  DX:      THE MINIMUM SEISMOMETER SPACING IN THE E-W
C                           DIRECTION
C                  DY:      THE MINIMUM SEISMOMETER SPACING IN THE N-S
C                           DIRECTION
C                  ALP(I):  COMPRESSIONAL WAVE VELOCITY IN LAYER I
C                  BET(I):  SHEAR WAVE VELOCITY IN LAYER I
C                  RHO(I):  DENSITY IN LAYER I
C                  Z(I):    THE DEPTH TO THE TOP OF LAYER I (Z INCREASES
C                           UPWARD)
C
C     LEN OF IWORK=6*MAX(NNW,NNP,NNQ)+150
C     LEN OF RWORK=6*MAX(NNW,NNP,NNQ)+150
C     LEN OF CWORK=MAX(NNP,NNQ)
C
C THIS VERSION (APRIL 25, 1987) USES ANALYTIC E MATRIX
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER IWORK(630)
      DIMENSION RWORK(630)
      COMPLEX*16 U1(40,20,20),U2(40,20,20),U3(40,20,20),
     !           CWORK(50),G(6),UT1(3),UT2(3),US1(3),US2(3)
      COMMON /DOMAIN/ W,P,Q
      COMMON /MEDIUM/ ALP(5),BET(5),RHO(5),Z(6)
      COMMON // UT1,UT2,DW,DP,DQ
      PI=3.14159265361
      READ(5,100)N,W0,DX,DY,(ALP(I),BET(I),RHO(I),Z(I),I=1,N)
100   FORMAT(I2,3F10.3/(4F10.3))
      NW=10
```

```
      NP=10
      NQ=10
      NWBLNK=NW/2
      NNW=2*NW+4*NWBLNK
      NNP=2*NP
      NNQ=2*NQ
      NW1=NWBLNK+1
      NWF=NWBLNK+NW
      IF (NWBLNK.EQ.0) DW=2.0*WO
      IF (NWBLNK.NE.0) DW=WO/NWBLNK
      PP=1.0/DX/2.0/PI
      QQ=1.0/DY/2.0/PI
      DP=PP/NNP
      DQ=QQ/NNQ
      DO 15 I=1,NNW
      DO 15 J=1,NNP
      DO 15 K=1,NNQ
        U1(I,J,K)=(0.0,0.0)
        U2(I,J,K)=(0.0,0.0)
15      U3(I,J,K)=(0.0,0.0)
      INDX1=NW+NWBLNK+1
      DO 10 I=NW1,NWF
         INDX1=INDX1-1
      WRITE(6,110)I,INDX1
110   FORMAT( 3I3,3E10.3)
        W=(I-0.5)*DW
        DO 10 J=1,NP
           P=(J-0.5)*DP-PP/2.0
             DO 10 K=1,NQ
               Q=(K-0.5)*DQ-QQ/2.0
               CALL SOURCE(N,G)
               IER=0
               CALL CALCU(N,G,UT1,IER)
C       WRITE(6,112)W,P,Q,UT1(3)
112     FORMAT('W,P,Q,UT1(3)',5E10.2)
               P=-P
               CALL SOURCE(N,G)
               CALL CALCU(N,G,UT2,IER)
               P=-P
               CALL FASSGN(U1,1,NNW,NNP,NNQ,INDX1,J,K)
```

```
                CALL FASSGN(U2,2,NNW,NNP,NNQ,INDX1,J,K)
10              CALL FASSGN(U3,3,NNW,NNP,NNQ,INDX1,J,K)
C       WRITE(6,111)(((U3(I,J,K),J=1,NNP),I=1,NNW),K=1,NNQ)
111     FORMAT('U3'/(8E10.2))
        CALL FFT3D(U1,40,20,NNW,NNP,NNQ,-1,IWORK,RWORK,CWORK)
        CALL FFT3D(U2,40,20,NNW,NNP,NNQ,-1,IWORK,RWORK,CWORK)
        CALL FFT3D(U3,40,20,NNW,NNP,NNQ,-1,IWORK,RWORK,CWORK)
        CALL CHSIGN(U1,NNW,NNP,NNQ)
        CALL CHSIGN(U2,NNW,NNP,NNQ)
        CALL CHSIGN(U3,NNW,NNP,NNQ)
C       WRITE(6,111)U3
        WRITE(7) U3
C       WRITE(6,111)(((U3(I,J,K),J=1,NNP),I=1,NNW),K=1,NNQ)
        STOP
        END
C
C ***************************************************************************
C
        SUBROUTINE SOURCE(N,G)
C
C *** THIS SUBROUTINE CALCULATES THE SOURCE CONTRIBUTION TO THE
C *** PROPAGATOR MATRIX ASSUMING THE SOURCE IS DELTA-LIKE IN
C *** SPACE AND TIME. THAT IS, IT CALCULATES G AS DEFINED BY
C *** EQUATIONS (2.31) AND (2.34).
C
C *** S   LAYER CONTAINING THE SOURCE (THE HALF-SPACE IS 1)
C *** G1   G* AS DEFINED BY (2.31)
C *** G2   G** AS DEFINED BY (2.31)
C *** N   THE NUMBER OF LAYERS (= INDEX OF TOP LAYER)
C *** M   THE SOURCE MOMENT TENSOR
C *** XS |
C *** YS | THE COORDINATES OF THE SOURCE LOCATION
C *** ZS |
C *** TS THE TIME OF THE SOURCE
C
        INTEGER N,S/2/
        REAL*8 W,P,Q,XS,YS,ZS,TS,ALP(5),BET(5),RHO(5),Z(6),M(3,3)
        COMPLEX*16 G(6),G1(6),G2(6),W1(6),W2(6),W3(6),II,A(6,6),
     !   PROP(6,6),EX
        COMMON /DOMAIN/ W,P,Q
```

```
      COMMON /MEDIUM/ ALP,BET,RHO,Z
C
C *** CALCULATE G* AND G** AS DEFINED BY EQUATION (2.21) ASSUMING
C *** XS=YS=0.0.
C
      ZS=25.0
      II=(0.0,1.0)
      DO 5 I=1,3
      DO 5 J=1,3
5        M(I,J)=0.0
      M(1,3)=1.0E10
      M(3,1)=1.0E10
      DO 10 I=1,3
         G1(I+3)=-II*(P*M(I,1)+Q*M(I,2))
         G1(I)=(0.0,0.0)
         G2(I+3)=M(I,3)
         G2(I)=(0.0,0.0)
10    CONTINUE
C
C *** CALCULATE THE PROPAGATOR MATRIX FROM THE TOP OF THE STACK
C *** OF LAYERS TO THE DEPTH OF THE SOURCE, AND MULT BY G.
C *** G=P*G1 +P*A*G2
C
      CALL PROP2(N,S,ZS,PROP)
      CALL MMULT(W1,6,1,PROP,6,G1)
      CALL MATRXA(ALP(S),BET(S),RHO(S),A)
      CALL MMULT(W2,6,1,A,6,G2)
      CALL MMULT(W3,6,1,PROP,6,W2)
      DO 20 I=1,6
20       G(I)=W1(I)+W3(I)
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE CALCU(N,G,U,IER)
C
C *** THIS SUBROUTINE CALCULATES THE TRANSFORM OF THE SURFACE
C *** DISPLACEMENT (U). SEE EQUATIONS (2.33) AND (2.36).
C
```

```
C *** N IS THE NUMBER OF LAYERS (INDEX OF TOP LAYER)
C *** G IS THE SOURCE TERM CALCULATED MY SUBROUTINE SOURCE
C *** U IS THE TRANSFORM OF THE SURFACE DISPLACEMENT
C
      INTEGER N,IER
      REAL*8 ALP(5),BET(5),RHO(5),Z(6),W,P,Q,RWORK(3)
      COMPLEX*16 E(6,6),PROP(6,6),B(3,3),C(3,3),CINV(3,3),G(6),
     !G1(3),G2(3),U(3),CWORK(15),W1(3,3),W2(3,3),W3(3,3),W4(3,3),ETA,ETB
      COMMON /DOMAIN/ W,P,Q
      COMMON /MEDIUM/ ALP,BET,RHO,Z
C
C *** FIND THE EIGENVECTOR MATRIX (E) FOR THE HALF-SPACE (Z=0)
C
      CALL EVECT1(ALP(1),BET(1),RHO(1),E)
C
C *** FIND THE PROPAGATOR MATRIX (PROP) FOR THE ENTIRE STACK OF LAYERS
C
      CALL PROP2(N,2,0.0D0,PROP)
C
C *** B=-(P11*E12 + P12*E22)
C *** C=(P21*E12 + P22*E22)
C *** CINV=C**-1
C *** SURFACE DISPLACEMENT=U=B*CINV*G2 + G1
C
      CALL SUBMAT(PROP,6,6,W1,3,3,1,3,1,3)
      CALL SUBMAT(E,6,6,W2,3,3,1,3,4,6)
      CALL MMULT(W3,3,3,W1,3,W2)
      CALL SUBMAT(PROP,6,6,W1,3,3,1,3,4,6)
      CALL SUBMAT(E,6,6,W2,3,3,4,6,4,6)
      CALL MMULT(W4,3,3,W1,3,W2)
      DO 10 I=1,3
         DO 10 J=1,3
10          B(I,J)=-(W3(I,J)+W4(I,J))
      CALL SUBMAT(PROP,6,6,W1,3,3,4,6,1,3)
      CALL SUBMAT(E,6,6,W2,3,3,1,3,4,6)
      CALL MMULT(W3,3,3,W1,3,W2)
      CALL SUBMAT(PROP,6,6,W1,3,3,4,6,4,6)
      CALL SUBMAT(E,6,6,W2,3,3,4,6,4,6)
      CALL MMULT(W4,3,3,W1,3,W2)
      DO 20 I=1,3
```

```
            DO 30 J=1,3
               CINV(I,J)=0.0
30          C(I,J)=W3(I,J)+W4(I,J)
            G1(I)=G(I)
            G2(I)=G(I+3)
20          CINV(I,I)=1.0
C
C *** USE IMSL ROUTINE LEQ2C TO INVERT B. BINV MUST BE THE IDENTITY
C *** MATRIX ON INPUT TO LEQ2C. PRINT W,P,Q AND RETURN IF LEQ2C
C *** CRASHES.
C
      CALL LEQ2C(C,3,3,CINV,3,3,0,CWORK,RWORK,IER)
      IF (IER.EQ.0) GOTO 50
         ETA=W**2/ALP(2)**2-P**2-Q**2
         ETA=CDSQRT(ETA)
         ETB=W**2/BET(2)**2-P**2-Q**2
         ETB=CDSQRT(ETB)
         WRITE(6,120)W,P,Q,ETA,ETB
         WRITE(6,130)((C(I,J),J=1,3),I=1,3)
130      FORMAT( 6E10.2)
120      FORMAT('*** CALCU** W,P,Q,ETA,ETB,C '/7D10.2)
         RETURN
50    CONTINUE
      CALL MMULT(W1,3,3,B,3,CINV)
      CALL MMULT(W2,3,1,W1,3,G2)
      DO 40 I=1,3
         U(I)=W2(I,1)+G1(I)
40    CONTINUE
C
      RETURN
      END
C
C ***************************************************************************
C
C *** GET A SUB MATRIX
C
      SUBROUTINE SUBMAT(A,NA,MA,B,NB,MB,IR,LR,IC,LC)
      COMPLEX*16 A(NA,MA),B(NB,MB)
      DO 1 I=IR,LR
      DO 1 J=IC,LC
```

```
1     B(I-IR+1,J-IC+1)=A(I,J)
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE MMULT(A,NA,MA,B,MB,C)
C
C *** MATRIX MULTIPLICATION SUBROUTINE
C *** A(NA,MA)=B(NA,MB)*C(MB,MA)
C
      COMPLEX*16 A(NA,MA),B(NA,MB),C(MB,MA)
      DO 1 I=1,NA
      DO 1 J=1,MA
      A(I,J)=0.0
      DO 1 K=1,MB
1     A(I,J)=A(I,J)+B(I,K)*C(K,J)
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE PROP2(TOP,BOTTOM,ZO,PROP)
C
C *** THIS SUBROUTINE CALCULATES THE PROPAGATOR MATRIX (PROP) FOR
C *** THE EARTH SECTION FROM THE TOP OF LAYER TOP TO ZO WHICH
C *** IS IN LAYER BOTTOM.
C
      INTEGER I,BOTTOM,TOP
      REAL*8 W,P,Q,ALP(5),BET(5),RHO(5),Z(6),ZZ(6),ZO
      COMPLEX*16 PRO(6,6),PROP(6,6),W1(6,6)
      COMMON /DOMAIN/W,P,Q
      COMMON /MEDIUM/ALP,BET,RHO,Z
C
      DO 10 I=1,6
         DO 15 J=1,6
           PROP(I,J)=(0.0,0.0)
15    CONTINUE
         PROP(I,I)=(1.0,0.0)
10    CONTINUE
```

```
C
C *** FOR THE ENTIRE THICKNESS OF EACH LAYER (WORKING FROM TOP TO
C *** BOTTOM), CALCULATE THE PROPAGATOR MATRIX (PRO). SINCE,
C ***       PROP(Z5,Z2)=PROP(Z5,Z3)*PROP(Z3,Z2)
C *** WHEN PRO IS FOR THE LAYER BETWEEN Z3 AND Z2, WE HAVE:
C ***       PROP=PROP*PRO
C *** WHERE PROP IS NOW FOR THE DEPTH Z5 TO Z2 INSTEAD OF THE
C *** PREVIOUS Z5 TO Z3. ON THE FIRST LOOP, PROP IS THE IDENTITY
C *** MATRIX, SO PROP(Z5,Z4) BECOMES PRO FOR THE ENTIRE 5TH LAYER.
C
      DO 25 I=1,TOP
25       ZZ(I)=Z(I)
      ZZ(1)=Z0
C     DO 20 I=TOP,BOTTOM,-1
      I=TOP+1
 7077 I=I-1
      IF (I.LT.BOTTOM) GOTO 20
         CALL PROPG2(ALP(I),BET(I),RHO(I),ZZ(I),ZZ(I-1),P,Q,W,PRO)
         CALL MMULT(W1,6,6,PROP,6,PRO)
         DO 30 L=1,6
         DO 30 M=1,6
30          PROP(L,M)=W1(L,M)
      GOTO 7077
20    CONTINUE
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE EVECT1(ALP,BET,RHO,E)
C
C *** THIS SUBROUTINE CALCULATES THE EIGENVECTOR MATRIX USING AN
C *** ANALYTIC DERIVATION (SEE CHAPTER 2).
C
      COMPLEX*16 I,ETA,ETB,E(6,6),WORK(6,6),WORK2(6,6),CWORK(48)
     !   ,EINV(6,6),A(6,6)
      REAL*8 W,P,Q,K,ALP,BET,RHO,LAM,MU,RWORK(84)
      COMMON /DOMAIN/ W,P,Q
      I=(0.0,1.0)
      K=DSQRT(P**2+Q**2)
```

```
ETA=W**2/ALP**2-K**2
ETA=CDSQRT(ETA)
ETB=W**2/BET**2-K**2
ETB=CDSQRT(ETB)
MU=RHO*BET**2
LAM=RHO*ALP**2-2*MU
E(1,1)=1*P
E(1,2)=1*P*ETB
E(1,3)=1*Q
E(1,4)=1*Q
E(1,5)=-1*P*ETB
E(1,6)=1*P
E(2,1)=1*Q
E(2,2)=1*Q*ETB
E(2,3)=-1*P
E(2,4)=-1*P
E(2,5)=-1*Q*ETB
E(2,6)=1*Q
E(3,1)=1*ETA
E(3,2)=-1*K**2
E(3,3)=0.0
E(3,4)=0.0
E(3,5)=-1*K**2
E(3,6)=-1*ETA
E(4,1)=-2*P*MU*ETA
E(4,2)=P*MU*(K**2-ETB**2)
E(4,3)=-Q*MU*ETB
E(4,4)=Q*MU*ETB
E(4,5)=P*MU*(K**2-ETB**2)
E(4,6)=2*P*MU*ETA
E(5,1)=-2*Q*MU*ETA
E(5,2)=Q*MU*(K**2-ETB**2)
E(5,3)=P*MU*ETB
E(5,4)=-P*MU*ETB
E(5,5)=Q*MU*(K**2-ETB**2)
E(5,6)=2*Q*MU*ETA
E(6,1)=(-2*MU*ETA**2*ALP**2-LAM*W**2)/ALP**2
E(6,2)=2*K**2*MU*ETB
E(6,3)=0.0
```

```
      E(6,4)=0.0
      E(6,5)=-2*K**2*MU*ETB
      E(6,6)=(-2*MU*ETA**2*ALP**2-LAM*W**2)/ALP**2
      RETURN
      END
C
C ***************************************************************
C
      SUBROUTINE MATRXA(ALP,BET,RHO,A)
C
C *** THIS SUBROUTINE CALCULATES THE MATRIX A BASED ON
C *** A OUTPUT FROM REDUCE2 FOR THE GIVEN SEISMIC VELOCITIES AND
C *** DENSITY. A IS DEFINED BY EQUATION (2.6).
C
      REAL*8 W,P,Q,ALP,BET,RHO,MEW,LAM
      COMPLEX*16 II,A(6,6)
      COMMON /DOMAIN/ W,P,Q
      II=(0.0,1.0)
C
C *** INITIALIZE A TO THE ZERO MATRIX
C
      DO 10 I=1,6
         DO 10 J=1,6
            A(I,J)=(0.0,0.0)
10    CONTINUE
C
      MEW=RHO*BET**2
      LAM=RHO*ALP**2-2*MEW
      A(1,3)=-II*P
      A(1,4)=1.0/(BET**2*RHO)
      A(2,3)=-II*Q
      A(2,5)=1.0/(BET**2*RHO)
      A(3,1)=(II*P*(2*BET**2-ALP**2))/ALP**2
      A(3,2)=(II*Q*(2*BET**2-ALP**2))/ALP**2
      A(3,6)=1.0/(RHO*ALP**2)
      A(4,1)=(RHO*(-W**2*ALP**2-4*P**2*BET**4+4*P**2*BET**2*ALP**2
     !+Q**2*BET**2*ALP**2))/ALP**2
      A(4,2)=(P*Q*BET**2*RHO*(-4*BET**2+3*ALP**2))/ALP**2
      A(4,6)=(II*P*(2*BET**2-ALP**2))/ALP**2
      A(5,1)=(P*Q*BET**2*RHO*(-4*BET**2+3*ALP**2))/ALP**2
```

```
      A(5,2)=(RHO*(-W**2*ALP**2+P**2*BET**2*ALP**2-4*Q**2*BET**4+4
     !*Q**2*BET**2*ALP**2))/ALP**2
      A(5,6)=(II*Q*(2*BET**2-ALP**2))/ALP**2
      A(6,3)=-W**2*RHO
      A(6,4)=-II*P
      A(6,5)=-II*Q
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE FASSGN(U,COORD,NI,NJ,NK,I,J,K)
C
C *** THIS SUBROUTINE FILLS OUT U, THE TRANSFORM OF THE SURFACE
C *** DISPLACEMENT, USING SYMMETRY RELATIONS. EG. U(P,Q,W)=
C *** U(-P,-Q,-W)*, WHERE * INDICATES COMPLEX CONJUGATE. SEE
C *** CHAPTER 3 FOR A COMPLETE DESCRIPTION.
C
      INTEGER NI,NJ,NK,I,J,K,A,B,C,COORD
      REAL*8 DW,DP,DQ
      COMPLEX*16 U(40,20,20),G1(3),G2(3)
      COMMON //G1,G2,DW,DP,DQ
C
C *** I,J, AND K CORRESPOND TO NEGATIVE FREQUENCIES AND WAVENUMBERS,
C *** A,B, AND C CORRESPOND TO POSITIVE FREQUENCIES AND WAVE-
C *** NUMBERS. THIS ROUTINE INIALIZES FOR A EVEN-NUMBERED FFT, AND
C *** THUS THE SPECTRUM IS NOT SAMPLED AT THE DC LEVEL.
C
      SCALE=DW*DP*DQ*NI*NJ*NK
      A=NI-I+1
      B=NJ-J+1
      C=NK-K+1
      U(I,J,K)=G1(COORD)*SCALE
      U(A,J,K)=U(I,J,K)
      U(I,B,K)=G2(COORD)*SCALE
      U(A,B,K)=U(I,B,K)
      U(I,J,C)=DCONJG(G2(COORD))*SCALE
      U(A,J,C)=U(I,J,C)
      U(I,B,C)=DCONJG(G1(COORD))*SCALE
      U(A,B,C)=U(I,B,C)
```

```
C
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE CHSIGN(U,NNW,NNP,NNQ)
C
C *** THIS SUBROUTINE APPLIES THE PHASE FACTOR NECESSARY FOR
C *** SMALL FFT'S (SEE EQUATION (3.6))
C
      INTEGER I,J,K
      REAL*8 EX,PI
      COMPLEX*16 U(40,20,20),II
      II=(0.0,1.0)
      PI=3.14159265359
      DO 10 I=1,NNW
      DO 10 J=1,NNP
      DO 10 K=1,NNQ
        EX=I-1.0-(I-1.0)/DFLOAT(NNW)+J-1.0-(J-1.0)/DFLOAT(NNP)
     !      +K-1.0-(K-1.0)/DFLOAT(NNQ)
        U(I,J,K)=CDEXP(PI*II*EX)*U(I,J,K)
10    CONTINUE
      RETURN
      END
```

When the source was descried by (2.31) and (4.14), the following subroutine was substituted for the one of the same name given above

```
      SUBROUTINE SOURCE(N,G)
C
C *** THIS SUBROUTINE CALCULATES THE SOURCE CONTRIBUTION TO THE
C *** PROPAGATOR MATRIX ASSUMING THE SOURCE IS GIVEN BY
C *** AAA*E**(-|X|/AA)*BB*E**(-|Y|/B)*D(T-TS)*D(Z-ZS)
C *** WHERE || DENOTES ABSOLUTE VALUE, AND D IS THE DIRAC DELTA.
C *** (EQUATION (4.12))
C
C *** S   LAYER CONTAINING THE SOURCE (THE HALF-SPACE IS 1)
C *** G1  G* AS DEFINED BY (2.23)
C *** G2  G** AS DEFINED BY (2.23)
```

```
C *** N   THE NUMBER OF LAYERS (= INDEX OF TOP LAYER)
C *** M   THE SOURCE MOMENT TENSOR
C *** XS |
C *** YS | THE COORDINATES OF THE SOURCE LOCATION
C *** ZS |
C *** TS THE TIME OF THE SOURCE
C
      INTEGER N,S/2/
      REAL*8 W,P,Q,XS,YS,ZS,TS,ALP(5),BET(5),RHO(5),Z(6),M(3,3)
     !       ,AA,AAA,B,BB
      COMPLEX*16 G(6),G1(6),G2(6),W1(6),W2(6),W3(6),
     !  II,A(6,6),PROP(6,6),EX
      COMMON /DOMAIN/ W,P,Q
      COMMON /MEDIUM/ ALP,BET,RHO,Z
C
C *** CALCULATE G* AND G** AS DEFINED BY EQUATIONS (2.32) AND (4.14)
C *** ASSUMING XS=YS=0.0
C
      ZS=25.0
      II=(0.0,1.0)
      AA=0.1
      AAA=1.0/2.0/AA
      B=0.1
      BB=1.0/2.0/B
      DO 5 I=1,3
      DO 5 J=1,3
5       M(I,J)=0.0
      M(1,3)=1.0E10
      M(3,1)=1.0E10
      DUMMY=AA**2*AAA*P/(1+AA**2*P**2) * B**2*BB*Q/(1+B**2*Q**2)
      DO 10 I=1,3
         G1(I+3)=-4*DUMMY*(M(I,1)/AA+M(I,2)/B)
         G1(I)=(0.0,0.0)
         G2(I+3)=M(I,3)
         G2(I)=(0.0,0.0)
10    CONTINUE
C
C *** CALCULATE THE PROPAGATOR MATRIX FROM THE TOP OF THE STACK
C *** OF LAYERS TO THE DEPTH OF THE SOURCE, AND MULT BY G.
C *** G=P*G1 +P*A*G2
```

```
C
      CALL PROP2(N,S,ZS,PROP)
      CALL MMULT(W1,6,1,PROP,6,G1)
      CALL MATRXA(ALP(S),BET(S),RHO(S),A)
      CALL MMULT(W2,6,1,A,6,G2)
      CALL MMULT(W3,6,1,PROP,6,W2)
      DO 20 I=1,6
20       G(I)=W1(I)+W3(I)
      RETURN
      END
```

When the source was described by (4.20) and (4.21), the following subroutine was substituted for the one of the same name given above.

```
      SUBROUTINE SOURCE(N,G)
C
C *** THIS SUBROUTINE CALCULATES THE SOURCE CONTRIBUTION TO THE
C *** PROPAGATOR MATRIX ASSUMING THE SOURCE IS GIVEN BY
C *** AAA*E**(-X**2/AA**2)*BB*E**(-Y**2/B**2)
C **     *CC*E**(-T**2/C**2)*D(Z-ZS)
C *** WHERE D IS THE DIRAC DELTA.
C *** (EQUATION (4.18))
C
C *** S  LAYER CONTAINING THE SOURCE (THE HALF-SPACE IS 1)
C *** G1 G* AS DEFINED BY (4.20)
C *** G2 G** AS DEFINED BY (4.21)
C *** N  THE NUMBER OF LAYERS (= INDEX OF TOP LAYER)
C *** M  THE SOURCE MOMENT TENSOR
C *** XS |
C *** YS | THE COORDINATES OF THE SOURCE LOCATION
C *** ZS |
C *** TS THE TIME OF THE SOURCE
C
      INTEGER N,S/2/
      REAL*8 W,P,Q,XS,YS,ZS,TS,ALP(5),BET(5),RHO(5),Z(6),M(3,3)
     !       ,AA,AAA,B,BB,PI
      COMPLEX*16 G(6),G1(6),G2(6),W1(6),W2(6),W3(6),
     !  II,A(6,6),PROP(6,6),EX,DUMMY
      COMMON /DOMAIN/ W,P,Q
      COMMON /MEDIUM/ ALP,BET,RHO,Z
```

```
C
C *** CALCULATE G* AND G** AS DEFINED BY EQUATIONS (4.20) AND
C *** (4.21) ASSUMING XS=YS=0.0
C
      PI=3.14159265359
      ZS=25.0
      II=(0.0,1.0)
      AA=0.1
      AAA=1.0/2.0/AA**2
      B=0.1
      BB=1.0/2.0/B**2
      C=0.1
      CC=1.0/2.0/C**2
      DO 5 I=1,3
      DO 5 J=1,3
5       M(I,J)=0.0
      M(1,3)=1.0E10
      M(3,1)=1.0E10
      DUMMY=AAA*BB*CC*AA*B*C*PI**(1.5)*
     ! DEXP(-0.25*(AA**2*P**2+B**2*Q**2+C**2*W**2))
      DO 10 I=1,3
        G1(I+3)=-II*AAA*P*DUMMY*M(I,1)-II*M(I,2)*BB*Q*DUMMY
        G1(I)=(0.0,0.0)
        G2(I+3)=M(I,3)*DUMMY
        G2(I)=(0.0,0.0)
10    CONTINUE
C
C *** CALCULATE THE PROPAGATOR MATRIX FROM THE TOP OF THE STACK
C *** OF LAYERS TO THE DEPTH OF THE SOURCE, AND MULT BY G.
C *** G=P*G1 +P*A*G2
C
      CALL PROP2(N,S,ZS,PROP)
      CALL MMULT(W1,6,1,PROP,6,G1)
      CALL MATRXA(ALP(S),BET(S),RHO(S),A)
      CALL MMULT(W2,6,1,A,6,G2)
      CALL MMULT(W3,6,1,PROP,6,W2)
      DO 20 I=1,6
20      G(I)=W1(I)+W3(I)
      RETURN
      END
```

When the Eigenvalue matrix was replaced by the matrix given in (4.26), the CALL to EVECT1 in soubroutine SOURCE was replaced by

```
CALL EVECT3(ALP(1),BET(1),RHO(1),E)
```

and the following subroutine was added to the program given above.

```
      SUBROUTINE EVECT3(ALP,BET,RHO,E)
C
C *** THIS SUBROUTINE CALCULATES THE E(W,P,Q)+E(W,-P,-Q)*
C *** WHERE E IS THE EIGENVECTOR MATRIX, AND * INDICATES
C *** COMPLEX CONJUGATE. THIS NEW MATRIX WILL TRANSFORM TO
C *** A REAL MATRIX. SEE CHAPTER 4 FOR DETAILS.
C
      COMPLEX*16 E(6,6),E1(6,6),E2(6,6)
      REAL*8 W,P,Q,K,ALP,BET,RHO
      COMMON /DOMAIN/ W,P,Q
      PP=P
      QQ=Q
      CALL EVECT1(PP,QQ,W,ALP,BET,RHO,E1)
      PP=-PP
      QQ=-QQ
      CALL EVECT1(PP,QQ,W,ALP,BET,RHO,E2)
      DO 10 L=1,6
      DO 10 M=1,6
10      E(L,M)=E1(L,M)+DCONJG(E2(L,M))
      RETURN
      END
```

Input/Output

An example of an input for the READ on unit 5 is:

```
2,30.,32.,24.,
2600.,1800.,2400.,0.,
2000.,1150.,2100.,225.,
```

This specifies a 1 layer over a half-space model. The first non-zero amplitude occurs at W0=30., and DX=32., DY=24. The output consists of a unformatted binary dump of the vertical surface displacement U3(t,x,y). This program is contained in file SYN, and Unit 5 is attached to SEISDATA on the University of Alberta MTS account ENY3.